

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 01-154242
(43)Date of publication of application : 16.06.1989

(51)Int.Cl. G06F 11/18
G06F 15/16

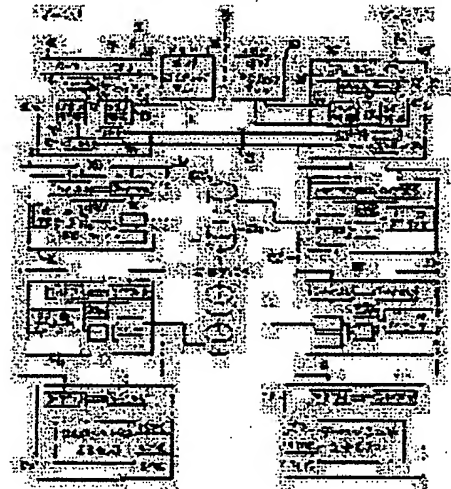
(21)Application number : 63-222174 (71)Applicant : DIGITAL EQUIP CORP <DEC>
(22)Date of filing : 05.09.1988 (72)Inventor : BRUCKERT WILLIAM F
BISSETT THOMAS D

(30)Priority
Priority number : 87 93572 Priority date : 04.09.1987 Priority country : US

(54) COMPUTER SYSTEM WITHSTANDING DEFECT WITH DUAL ZONES

(57)Abstract:

PURPOSE: To configure the computer system withstanding defect with dual zones operated usually simultaneously by providing 1st and 2nd clock means connecting respectively to 1st and 2nd data processors.
CONSTITUTION: Since the system withstanding defect 10 has duplicate components, no single defect is caused. Processing systems 20, 20' are defect stop processing systems and when each of the systems detects a defect or an error in its sub system, the propagation of the defect or error to other sub systems without being controlled. The two defect stop processing systems 20, 20' are interconnected by some components in operation in a specified state act like a fail safe system and even when either of the defect stop processing systems 20, 20' is failed, the processing is continued as the entire computer system.



⑫ 公開特許公報(A) 平1-154242

⑬ Int. Cl.⁴G 06 F 11/18
15/16

識別記号

3 1 0
4 7 0

庁内整理番号

E-7368-5B
J-6745-5B

⑭ 公開 平成1年(1989)6月16日

審査請求 未請求 請求項の数 11 (全55頁)

⑮ 発明の名称 二重ゾーンの耐欠陥コンピュータシステム

⑯ 特 願 昭63-222174

⑰ 出 願 昭63(1988)9月5日

優先権主張 ⑱ 1987年9月4日 ⑲ 米国(U S) ⑳ 093572

㉑ 発 明 者 ウィリアム エフ ブ アメリカ合衆国 マサチューセッツ州 01532 ノースボ
ルツカート ロ マツシュビー サークル 13㉒ 発 明 者 トーマス デイー ビ アメリカ合衆国 ニューハンプシャー州 03038 デリー
セツト オルセン ロード 21㉓ 出 願 人 デジタル イクイブ アメリカ合衆国 マサチューセッツ州 01754 メイナ
メント コーポレーシ ド メイン ストリート 146番
ヨン

㉔ 代 理 人 弁理士 中 村 稔 外7名

明 細 書

1. 発明の名称

二重ゾーンの耐欠陥コンピュータシステム

2. 特許請求の範囲

(1) 第1の処理システムを具備し、このシステムは、一連のデータ処理命令を実行するための第1データプロセッサと、上記第1の処理システムからデータを出力するための第1データ出力ターミナルとを備えており、

更に、上記第1の処理システムと実質的に同じで且つ上記第1の処理システムとは独立して動作する第2の処理システムを具備し、この第2の処理システムは、上記第1データプロセッサと同じシーケンスで上記一連のデータ処理命令を実行するための第2データプロセッサと、上記第2の処理システムからデータを出力するための第2のデータ出力ターミナルとを備えており、

更に、上記第1及び第2のデータプロセッサに接続され、上記第1及び第2の処理システムによる上記一連のデータ処理命令の実行を同期状態

に維持するための同期手段を具備し、

更に、上記第1及び第2のデータ出力ターミナルに接続されて、上記第1の処理システムから出力されたデータを上記第2の処理システムから出力されたデータと比較するための欠陥検出手段を具備し、この欠陥検出手段は、上記第1の処理システムから上記第1の出力ターミナルに出力された上記データが上記処理システムから上記第2の出力ターミナルに出力されたデータと異なるときにエラーの存在を識別する手段を備えていることを特徴とする耐欠陥コンピュータシステム。

(2) 上記同期手段は、

上記第1のデータプロセッサに接続されて、上記第1のデータプロセッサのためのタイミング信号を発生する第1クロック手段と、

上記データプロセッサに接続されて、上記第2のデータプロセッサのためのタイミング信号を発生する第2のクロック手段とを備えた請求項1に記載の耐欠陥コンピュータシステム。

(3) 上記第1のクロック手段は、

上記第1のデータ処理システムのための上記第1のタイミング信号を発生する第1発振器手段と、

上記第1発振器手段に接続されて、上記第1及び第2のタイミング信号を同期させる位相ロック手段とを備えており、そして上記第2のクロック手段は、

上記第2のデータ処理システムのための上記第2のタイミング信号を発生する第2発振器手段と、

上記第2発振器手段に接続されて、上記第2及び第1のタイミング信号を同期させる第2の位相ロック手段とを備えた請求項2に記載の耐欠陥コンピュータシステム。

(4) 更に、上記第1及び第2のデータ出力ターミナルに接続され、上記第1及び第2の処理システムからの上記データ出力を受け取る第1出力モジュールを具備し、この第1出力モジュールは、上記第1及び第2の処理システムからのデータ出力を上記第1の出力モジュールにおいて更に伝達

(5) 上記第1及び第2のデータ比較手段は、各々、上記第1及び第2の処理システムからの上記データ出力が異なるときに上記第1及び第2の処理システムにエラー信号を送信するエラー信号発生手段を備えている請求項4に記載の耐欠陥コンピュータシステム。

(6) 上記第1と第2の処理システム間に両方向性のデータ経路を形成するクロスリンク通信手段を更に備えた請求項1に記載の耐欠陥コンピュータシステム。

(7) 上記通信手段は、第1及び第2のクロスリンク要素を一緒に備えており、第1のクロスリンク要素は、

上記第1の処理システムから上記第1のデータ出力ターミナル及び上記第2の処理システムへデータ出力を転送する第1出力ルート手段と、

上記第2の処理システムから上記第1の出力モジュールへデータ出力を転送する第1入力ルート手段とを含み、そして更に、上記第2クロスリンク要素は、

する前にそれらが等しいかどうかチェックするための第1データ比較手段と、この第1データ比較手段に接続されて、上記第1及び第2の処理システムからのデータ出力を合流させて上記第1出力モジュールのための第1出力流に形成する第1データ合流手段とを備えており、そして

上記第1及び第2のデータ出力ターミナルに接続され、上記第1及び第2の処理システムからの上記データ出力を受け取る第2出力モジュールを具備し、この第2出力モジュールは、上記第1及び第2の処理システムからのデータ出力を上記第2の出力モジュールにおいて更に伝達する前にそれらが等しいかどうかチェックするための第2データ比較手段と、上記第1及び第2の処理システムからのデータ出力を合流させて上記第2出力モジュールのための第2出力流に形成する第2データ合流手段とを備え、そして更に、上記欠陥検出手段の上記識別手段は、上記第1及び第2のデータ比較手段を含む請求項1に記載の耐欠陥コンピュータシステム。

上記第2のデータ処理システムから上記第1の処理システムへデータ出力を転送する第2出力ルート手段と、

上記第1の処理システムから上記第2の出力モジュールへデータ出力を転送する第2入力ルート手段とを含んでいる請求項6に記載の耐欠陥コンピュータシステム。

(8) 上記第1及び第2のデータプロセッサは第1及び第2のメモリを備え、そして各々、

上記第1出力手段は、上記第1メモリユニットから上記第1データ出力ターミナル及び上記第2の処理システムへデータを転送するための手段を備えており、そして

上記第2出力手段は、上記第2メモリユニットから上記第2データ出力ターミナル及び上記第1の処理システムへデータを転送するための手段を備えている請求項7に記載の耐欠陥コンピュータシステム。

(9) 上記第1及び第2のデータプロセッサは第1及び第2対の中央処理ユニットを各々含み、

上記第1データプロセッサにおける上記第1対の中央処理ユニットは上記第1メモリユニットに接続されそして上記第2データプロセッサにおける第2対の中央処理ユニットは上記第2メモリユニットに接続される請求項8に記載の耐欠陥コンピュータシステム。

(10) 上記識別手段によるエラー識別に対応して、そのエラーを生じた耐欠陥コンピュータシステムの部分を探索するためのエラー分離手段を更に備えた請求項1に記載の耐欠陥コンピュータシステム。

(11) 上記エラー分離手段に接続されて、エラーを生じた上記コンピュータシステムの要素を動作から取り去る回復手段を更に備えた請求項10に記載の耐欠陥コンピュータシステム。

タスクを実行する。更に別の解決策は、各々それ自身のメモリを有する2つ以上のマイクロプロセッサを使用してこれらが同じタスクを一緒に行うようにすることである。更に別の解決策は、共通のメモリを共有する2つ以上のマルチプロセッサを用いて同じタスクを一緒に行うことである。

上記の4つの解決策を組み合わせて使用する耐欠陥コンピュータシステムも考えられる。1つの公知の耐欠陥コンピュータシステムにおいては、4つの中央処理ユニットが使用され、そのうちの2つは第1のボードにおいてデュプレックス形態で動作しそして他の2つは第2のボードにおいてデュプレックス形態で動作する。各ボードからの出力が等しいとき及び等しくないときを検出するために比較器が使用される。これらが等しくないことが比較器によって検出された場合には、欠陥ボードが自動的に停止され、その後、他のボードによって出力が与えられる。

発明が解決しようとする課題

公知の耐欠陥コンピュータシステムは、種々

3. 発明の詳細な説明

産業上の利用分野

本発明は、耐欠陥マルチプロセッサコンピュータシステムを形成する方法及びシステムに係る。

従来の技術

データ処理が中断されないことが重要であるようなコンピュータシステムの用途は多数ある。このような用途としては、例えば、ファイナンスの業界、原子力プラントのような厳密な工業設備、そして一般には、コンピュータシステムの欠陥によって重大な破壊を招く場合が挙げられる。

耐欠陥コンピュータシステムは、或る欠陥が生じてデータ処理を継続できるように複数のシステム又はシステム部品を備えた色々な程度の冗長度で形成されている。耐欠陥コンピュータシステムを得るために多数の解決策が利用されている。1つの解決策においては、各々がそれ自身のメモリを有した多数のマルチプロセッサが別々のタスクを実行する。別の解決策においては、多数のマルチプロセッサが共通のメモリを共有し、別々の

の程度の耐欠陥特性を与えるが、その本来の目的を満たさず、然も、本発明によってもたらされるような効果も与えない。

そこで、本発明の目的は、通常同時に動作する二重のコンピュータシステムを有した耐欠陥コンピュータ方法及びシステムを提供することである。この二重構成は、単一欠陥点が生じないように確保し、一方のシステムのエラー又は欠陥によって全コンピュータシステムが作動不能にならないようにする。更に、このような欠陥は、全て、エラーを生じさせたモジュール又は素子を作動不能にするか又は無視することによって修正できる。

本発明の更に別の目的及び効果は、以下の説明にその一部分が記載されていると共に、以下の説明からその一部分が明らかであろうし、又、本発明を実施することによって学び取ることができよう。本発明の目的及び効果は、特許請求の範囲に特に指摘する方法、装置及びその組み合わせによって実現されそして達成されるであろう。

課題を解決するための手段

これら及び他の目的を達成するために、以下で述べるように実施される本発明によれば、第1の処理システムを具備し、このシステムは、一連のデータ処理命令を実行するための第1データプロセッサと、上記第1の処理システムからデータを出力するための第1データ出力ターミナルとを備えており、更に、上記第1の処理システムと実質的に同じで且つ上記第1の処理システムとは独立して動作する第2の処理システムを具備し、この第2の処理システムは、上記第1データプロセッサと同じシーケンスで上記一連のデータ処理命令を実行するための第2データプロセッサと、上記第2の処理システムからデータを出力するための第2のデータ出力ターミナルとを備えており、更に、上記第1及び第2のデータプロセッサに接続され、上記第1及び第2の処理システムによる上記一連のデータ処理命令の実行を同期状態に維持するための同期手段を具備し、更に、上記第1及び第2のデータ出力ターミナルに接続されて、上記第1の処理システムから出力されたデータを

位相ロック手段とを備えており、そして上記第2のクロック手段は、上記第2のデータ処理システムのための上記第2のタイミング信号を発生する第2発振器手段と、上記第2発振器手段に接続されて、上記第2及び第1のタイミング信号を同期させる第2の位相ロック手段とを備えている。

本発明の更に別の特徴において、上記第1及び第2のデータ出力ターミナルに接続され、上記第1及び第2の処理システムからの上記データ出力を受け取る第1出力モジュールを具備し、この第1出力モジュールは、上記第1及び第2の処理システムからのデータ出力を上記第1の出力モジュールにおいて更に伝達する前にそれらが等しいかどうかチェックするための第1データ比較手段と、この第1データ比較手段に接続されて、上記第1及び第2の処理システムからのデータ出力を合流させて上記第1出力モジュールのための第1出力流に形成する第1データ合流手段とを備えており、そして上記第1及び第2のデータ出力ターミナルに接続され、上記第1及び第2の処理シス

上記第2の処理システムから出力されたデータと比較するための欠陥検出手段を具備し、この欠陥検出手段は、上記第1の処理システムから上記第1の出力ターミナルに出力された上記データが上記第2処理システムから上記第2の出力ターミナルに出力されたデータと異なるときにエラーの存在を識別する手段を備えていることを特徴とする耐欠陥コンピュータシステムが提供される。

本発明の1つの特徴において、上記同期手段は、上記第1のデータプロセッサに接続されて、上記第1のデータプロセッサのためのタイミング信号を発生する第1クロック手段と、上記データプロセッサに接続されて、上記第2のデータプロセッサのためのタイミング信号を発生する第2のクロック手段とを備えている。

本発明の別の特徴において、上記第1のクロック手段は、上記第1のデータ処理システムのための上記第1のタイミング信号を発生する第1発振器手段と、この第1発振器手段に接続されて、上記第1及び第2のタイミング信号を同期させる

テムからの上記データ出力を受け取る第2出力モジュールを具備し、この第2出力モジュールは、上記第1及び第2の処理システムからのデータ出力を上記第2の出力モジュールにおいて更に伝達する前にそれらが等しいかどうかチェックするための第2データ比較手段と、上記第1及び第2の処理システムからのデータ出力を合流させて上記第2出力モジュールのための第2出力流に形成する第2データ合流手段とを備え、そして更に、上記欠陥検出手段の上記識別手段は、上記第1及び第2のデータ比較手段を含む。

本発明の更に別の特徴において、上記第1及び第2のデータ比較手段は、各々、上記第1及び第2の処理システムからの上記データ出力が異なるとき上記第1及び第2の処理システムにエラー信号を送信するエラー信号発生手段を備えている。

本発明の他の特徴において、上記第1と第2の処理システム間に両方向性のデータ経路を形成するクロスリンク通信手段を更に備え、このクロスリンク通信手段は、一緒に接続された第1及び

第2のクロスリンク要素を備えており、第1のクロスリンク要素は、上記第1の処理システムから上記第1のデータ出力ターミナル及び上記第2の処理システムへデータ出力を転送する第1出力ルート手段と、上記第2の処理システムから上記第1の出力モジュールへデータ出力を転送する第1入力ルート手段とを含み、更に、上記第2クロスリンク要素は、上記第2のデータ処理システムから上記第1の処理システムへデータ出力を転送する第2出力ルート手段と、上記第1の処理システムから上記第2の出力モジュールへデータ出力を転送する第2入力ルート手段とを含んでいる。

本発明の別の特徴によれば、上記第1及び第2のデータプロセッサは第1及び第2のメモリを備え、そして各々、上記第1出力手段は、上記第1メモリユニットから上記第1データ出力ターミナル及び上記第2の処理システムへデータを転送するための手段を備えておりそして上記第2出力手段は、上記第2メモリユニットから上記第2データ出力ターミナル及び上記第1の処理システム

へデータを転送するための手段を備えている。

本発明の更に別の特徴によれば、上記第1及び第2のデータプロセッサは第1及び第2対の中央処理ユニットを各々含み、上記第1データプロセッサにおける上記第1対の中央処理ユニットは上記第1メモリユニットに接続されそして上記第2データプロセッサにおける第2対の中央処理ユニットは上記第2メモリユニットに接続される。

本発明の別の特徴において、耐欠陥コンピュータシステムは、上記識別手段によるエラー識別にตอบสนองして、そのエラーを生じた耐欠陥コンピュータシステムの部分を探索するためのエラー分離手段を備えている。

本発明の別の特徴において、耐欠陥コンピュータシステムは、上記エラー分離手段に接続されて、エラーを生じた上記コンピュータシステムの要素を動作から取り去る回復手段を備えている。

本明細書に含まれてその一部分を構成する添付図面は、本発明の一実施例を示すもので、これを参照しながら本発明の原理を説明する。

実施例

以下、添付図面を参照し、本発明の好ましい実施例を詳細に説明する。

A. システムの説明

第1図は、本発明の目的を達成する耐欠陥コンピュータシステム10のブロック図である。この耐欠陥コンピュータシステム10は、ゾーン又はステーションと称する二重のシステムを含んでいる。通常のモードにおいては、これらゾーンが同時に作動する。この二重構成では、単一欠陥点が生じないように確保すると共に、一方のゾーンのエラー又は欠陥によってコンピュータシステム10が作動不能にならないようにする。更に、このような全ての欠陥は、その欠陥を生じたモジュール又は素子を作動不能にするか又は無視することによって修正することができる。2つのゾーン11及び11'が二重の処理システム20及び20'を含むものとして第1図に示されている。然し乍ら、二重性は処理システムの範囲を越える。

第2図は、耐欠陥コンピュータシステム10

の物理的なハードウェアを示していると共にシステムの二重性を示している。各ゾーン11及び11'は、各々、別々のキャビネット12及び12'に収容される。キャビネット12は、バッテリー13と、電力レギュレータ14と、冷却ファン16と、交流入力17とを備えている。キャビネット12'は、キャビネット12の要素12-14、16及び17に対応する別々の要素を備えている。

以下で詳細に述べるように、処理システム20及び20'は、バックプレーンによって相互接続された多数のモジュールを備えている。或るモジュールが欠陥又はエラーを含んでいる場合には、コンピュータシステム10を作動不能にすることなくそのモジュールを取外して交換することができる。これは、処理システム20及び20'が物理的に別々のものであって、別々のバックプレーンを有しており、これらバックプレーンにモジュールを差し込んで互いに独立して動作させることができるからである。従って、一方の処理システムが動作を続けている間に他方の処理システムの

バックプレーンからモジュールを取り外したり差し込んだりすることができる。

二重の処理システム 20 及び 20' は同一のものであって、同一のモジュールを含んでいる。従って、処理システム 20 についてのみ完全に説明すれば、処理システム 20' が同等に動作することが理解されよう。

処理システム 20 は、第 3 図及び第 4 図に詳細に示された CPU モジュール 30 を備えている。この CPU モジュール 30 は、以下で詳細に説明するクロスリンク通路 25 によって処理システム 20' の CPU モジュール 30' に相互接続される。クロスリンク通路 25 は、処理システム 20 と 20' との間のデータ送信経路を形成し、処理システム 20 及び 20' が同期して動作するように確保するタイミング信号を搬送する。

又、処理システム 20 は、第 3 図及び第 17 図に詳細に示された I/O モジュール 100、110 及び 120 を備えている。これら I/O モジュール 100、110 及び 120 の各々は、二重

一般に、2つの欠陥停止処理システム 20 及び 20' はロックステップ同期状態で動作する。3つの重要な例外がある。その第 1 は、以下で詳細に述べるブートストラップ技術によって両方のプロセッサを同期状態にもっていくときの初期化である。第 2 の例外は、処理システム 20 及び 20' が 2 つの異なるワークロードにおいて独立して（非同期で）動作するときである。第 3 の例外は、処理システム 20 及び 20' に幾つかのエラーが生じるときである。この第 3 の例外においては、処理システム又はモジュールの 1 つが動作不能にされ、同期動作が終了となる。

二重処理システム 20 及び 20' の同期は、同じ入力を受けて同じ既知の状態からスタートするとき常に同じマシン状態に入ってエラーが生じない限り同じ結果を与える決定的なマシンとして各システムを処理することによって実行される。処理システム 20 及び 20' は、同一の構成にされ、同じ入力を受け、それ故、同じ状態を通る。従って、両方のプロセッサが同期して動作する限

り、ルールモジュール相互接続部 130 及び 132 によって CPU モジュール 30 に接続されている。モジュール相互接続部 130 及び 132 は、処理システム 20 のためのバックプレーンとして働く。

B. 耐欠陥システムの原理

耐欠陥コンピュータシステム 10 は、各要素が二重になっているので、単一欠陥点をもつことがない。処理システム 20 及び 20' の各々は、欠陥停止処理システムであり、即ち、これらシステムはサブシステム内の欠陥又はエラーを検出するとそれらが制御されない状態で他のサブシステムへ伝播するのを防止することができる。

2 つの欠陥停止処理システム 20 及び 20' は、規定の状態で動作する幾つかの要素によって相互接続されて、フェイルセーフシステムを形成する。耐欠陥コンピュータシステム 10 として実施されるフェイルセーフシステムにおいては、欠陥停止処理システム 20 及び 20' の一方に欠陥が生じた場合でもコンピュータシステム全体が処理を続けることができる。

りこれらプロセッサが同じ結果を与えそして同じ状態に入らなければならない。処理システムが同じ状態にないか又は異なった結果を与える場合には、処理システム 20 及び 20' の一方に欠陥が生じたと仮定される。修正処置をとるためには欠陥源を分離し、例えば、欠陥モジュールを動作不能にしなければならない。

エラー検出には、一般に、付加的な処理時間又は論理の形態のオーバーヘッドが含まれる。このようなオーバーヘッドを最小にするためには、耐欠陥動作に適合するようにできるだけ頻繁にならないようにシステムがエラーをチェックしなければならない。CPU モジュール 30 及び 30' からデータが出力される前に行うエラーチェックは非常に僅かである。さもなくば、原子炉の場合と同様に内部の処理エラーによって外部のシステムに不適切な動作が生じ、このような状態を防止するように耐欠陥システムが設計されている。

付加的なエラーチェックを行う理由は幾つか

ある。例えば、欠陥又はエラーを分離するためには、CPUモジュール30及び30'によって受け取ったデータを記憶又は使用の前にチェックすることが望ましい。さもなくば、エラーのある記憶データが後でアクセスされて付加的なエラーが生じたときには、エラーの発生源を見つけることが困難もしくは不可能となる。これは、特に、エラーのあるデータがいつの間にか記憶された場合にいえることである。時間がたったりエラーデータが後で処理されたときにはエラー発生源への追跡動作が破壊される。

エラーが検出される前に記憶される時間の長さを表わす「エラー待ち時間」も後で問題を生じさせる。例えば、めったに使用しないルーチンは、コンピュータシステムが手前のエラーによって減少した容量で既に動作しているときには待ち時間エラーをカバーしない。コンピュータシステムの容量が減少したときには、待ち時間エラーによってシステムがクラッシュする。

更に、二重レールシステムである処理システ

ムは、当業者に知られている標準的な中央処理ユニットである。ここに述べる好ましい実施例では、CPU40及び50は、本発明の譲受人であるデジタルエクイップメント社によって製造されたVAX（登録商標）プロセッサである。

CPU40及び50には、これらCPUに対して十分なメモリサイズの標準的なキャッシュRAMであるキャッシュメモリ42及び52が各々組み合わされる。好ましい実施例においては、キャッシュRAMが4K×64ビットである。然し、本発明は、キャッシュRAMをもつ必要はない。

CPU40とキャッシュ42との間のインターフェイスとして働くのは、システムサポート・キャッシュ制御要素44であり、CPU50とキャッシュ52との間のインターフェイスとして働くのは、システムサポート・キャッシュ制御要素54である。要素44と54は同一であり、その各々は対応するキャッシュとCPUとの間の標準的なインターフェイスをなし、又、インターバルタイマのような従来の周辺装置機能をCPUに与

える。メモリ20及び20'においては、メモリのような共有リソースのごとき単一レールシステムにデータを転送する前にエラーをチェックすることが所望される。というのは、このような転送の後には2つの独立したデータリソースがもはやなく、単一レールシステムのエラーが後で検出された場合には、エラーの追跡が不可能ではないまでも困難になるからである。

C. モジュールの説明

1. CPUモジュール

第1図に示されたCPUモジュール30の要素が第3図及び第4図に詳細に示されている。第3図はCPUモジュールのブロック図であり、第4図はCPUモジュール30、I/Oモジュール100及びそれらの相互接続部のブロック図である。CPUモジュール30及び30'に含まれた要素及びその動作は同じであるから、CPUモジュール30についてのみ説明する。

CPUモジュール30は、二重のCPU40及び50を含んでいる。これらCPU40及び50

は、キャッシュバス43及び53は、CPU40及び50を各々システムサポート・キャッシュ制御モジュール52及び42に接続する。

2. メモリモジュール

CPU40及び50は、4つまでのメモリモジュール60を共有できるのが好ましい。第5図は1つのメモリモジュール60のブロック図であり、そして第6図はモジュール60の特定のメモリ要素を示す詳細図である。

メモリモジュール60は、一次メモリ制御器70から32ビット両方向メモリバス85を経てデータを受け取る。又、メモリモジュール60は、メモリ制御器70及び75から各々バス80及び82を経てアドレス/制御信号も受け取る。バス80及び82は、行及び列のアドレス信号と、タイミング及び制御信号、例えば、RAS（行アドレスストロブ）、CAS（列アドレスストロブ）、WE（書き込みイネーブル）及びリフレッシュ信号とを含んでいる。

第5図に示すように、メモリモジュール60

はメモリアレイ600を含んでいる。このメモリアレイ600は、行及び列アドレスによってアドレスできる標準的なRAMであるのが好ましい。好ましい実施例では、メモリアレイ600は、メモリを8バンクまで含むことができる。

制御論理回路610は、アドレス及びメモリ制御信号を伝送しそしてタイミング及び内部制御信号を発生する。第6図に詳細に示されたように、制御論理回路610は、一次制御信号デマルチプレクサ612と、ミラー制御信号デマルチプレクサ614と、一次制御信号ドライバ616と、システムタイミング制御信号発生器618とを備えている。一次制御信号デマルチプレクサ612は、一次ボードアドレス及び一次バンクアドレス信号及び一次RAS、CAS、WE及びリフレッシュ信号を一次メモリ制御器70から受け取る。一次ボードアドレス信号が特定のボードを識別する場合には、そのボードがアクティブとなり、そのボードのデマルチプレクサ612は、RAS、CAS、リフレッシュ及び3ビットのバンクアドレス

信号から2組の8個の制御信号を発生する。これら8個の制御信号は、一次制御信号ドライバ616からの8個のPRAS（一次RAS）及び8個のPCAS（一次CAS）信号となる。ドライバ616は、これらの信号をブーストして種々のメモリバンクを駆動する。アクティブなメモリボードについては、デマルチプレクサ612は、WE信号をデマルチプレクスしてその4つのコピーを形成し、これらはドライバ616によってブーストされて4つのPWE（一次WE）信号が形成される。リフレッシュ動作中には、全てのボードがアクティブとなる。

ミラー制御信号デマルチプレクサ614は、ミラー信号に対して等しく動作する。ミラー信号は別々のボードに供給されず、エラー検出に使用されるだけであるから、ドライバを必要としない。

システムタイミング/制御信号発生器618は、4種類の入力、即ちクロック信号と、書き込み、読み取り及びリフレッシュタイミングのようなメモリサイクル信号と、当業者に良く知られた幾つ

かの他のシステム制御信号と、アドレスビット29とを受け取る。アドレスビット29は、アドレス信号がメモリスペース（即ち、メモリアレイ600）へのアクセスを識別するか又はI/Oスペース（I/O装置又はシステムレジスタの1つ）へのアクセスを識別するかを決定する。システムタイミング制御信号発生器618は、以下に述べるメモリモジュール60の他の要素の整合及びタイミングを制御する。

第5図に示された構成エラー論理回路620は、メモリモジュール60の動作中に検出されたエラーに関連した情報を記憶する。特に、比較論理回路630によってエラーが検出された場合には、構成エラー論理回路620は、その欠陥アドレス及び/又はデータを識別するに必要な情報を記憶する。然し乍ら、比較論理回路630は、制御及びアドレス信号をチェックするだけで、メモリデータ信号はチェックしない。

メモリデータ信号はエラー検出コード（EDC）を用いてチェックされる。好ましい実施例で

は、メモリ制御器70及び75によって必要とされるものと同一コードを使用し、これは、単一ビット修正、二重ビット検出、エラー修正コード（ECC）であるのが好ましい。

第6図に示されたように、構成エラー論理回路620は、エラー処理論理回路625及びEEPROM626を備えている。エラー処理論理回路625は、エラーカウンタと、制御論理回路と、4つの記憶レジスタ（1つは一次アドレス用、1つは二次アドレス用、1つはECC用そして1つはデータワード用）とを備えている。論理回路625は、以下で詳細に述べる比較論理回路630の出力からエラー信号を発生する。好ましくは、エラー状態が検出されたときに、カウンタが増加し、論理回路625のレジスタが一次及びそのミラーメモリアドレスと、ECCと、それに関連したデータワードとを記憶する。何等かの形式のNVRAM（不揮発性RAM）であるEEPROM626は、オフライン診断に対してメモリエラーデータを記憶する。メモリモジュールに欠陥が生

じた後にこのモジュールが取り外されたときには、欠陥の原因を判断するためにEEPROM 626から記憶されたデータが取り出される。

比較論理回路630は、一次メモリ制御器70からのECC、制御及びアドレス信号をミラーメモリ制御器75からの信号と比較することによりエラー検出を行い、これら信号が互いに等しくないことを検出する。比較論理回路630は、第6図に詳細に示されており、行及び列アドレスメモリドライバ632、アドレス比較器634、制御信号比較器636、ECC信号比較器638、データ及びECCトランシーバ640、及びECC発生器642を備えている。

行及び列アドレスメモリドライバ632は、11ビットの行及び列アドレス信号を受け取って各信号の4つのコピーを発生し、メモリアレイ600の全てのメモリバンクに対して十分な信号強度を与える。

各々の一次行及び列アドレス信号の4つのコピーは、ミラー行及び列アドレス信号と同様に、

にするためにアドレス比較器634のタイミングを制御する。比較の結果は、エラー処理及び制御論理回路625へ入力される。アドレス比較器634がいずれかの対応する信号が互いに異なることが分かった場合には、回路625がエラーを指示し、アドレス/制御エラー信号762をメモリ制御器70及び80に送ることによって適当な処置をとる。

制御信号比較器636は、アドレス信号比較器634と同様に動作する。制御信号比較器636は、PRAS信号とMRAS信号を比較し、PCAS信号とMCAS信号を比較し、PWE信号とMWE信号を比較しそして一次リフレッシュ信号とミラーリフレッシュ信号を比較する。制御信号比較器636は、アドレス比較器634と同様にタイミングどりにされ、比較エラーが指示されたときに、エラー処理及び制御論理回路625はエラーを指示し、アドレス/制御エラー信号をメモリ制御器70及び75に送信する。

ECC比較回路638は、比較器634及び

アドレス比較器634へ入力される。メモリモジュール60の好ましい実施例においては、行及び列の両方のアドレスが11ビットの長さであり、交互のサイクルにバス80及び82を経て送信される。従って、各メモリアドレスごとに、2つの比較が順次行われる。

アドレス比較器634は、別々の信号に各々対応する11個の5入力排他的オアゲート回路を用いて44個の一次行及び列アドレス信号と11個のミラー行及び列アドレス信号を同時に比較する。排他的オアゲート回路は、いずれかの入力が増える場合にそれらの出力をイネーブルする。同様に、アドレス比較器634は、一次ボードアドレス信号とミラーボードアドレス信号を比較すると共に、一次バンクアドレス信号とミラーバンクアドレス信号を比較する。これら全ての信号は、デマルチプレクサ612及び614へも入力される。

システムタイミング及び制御信号発生器618は、入力信号が安定したときに比較を行うよう

636とは若干異なった動作をする。書き込み動作の場合、ECC比較器638は、データ及びECCトランシーバ640から7ビットの一次ECCデータを受け取る。トランシーバ640はメモリアレイ600に対するデータ及びECC信号をバッファする。次いで、ECC比較器638は、トランシーバ640からのECC信号と、トランシーバ640の出力の32ビット一次データ信号からECC発生器642によって形成されたECC信号とを比較する。

又、ECC比較器638は、ミラーメモリ制御器75から受け取ったミラーECC信号と、一次メモリ制御器70から受け取った一次ECC信号とを比較する。いずれかのECC比較によって2つの信号が等しくないことが指示された場合には、エラー処理及び制御論理回路625がエラーを指示し、ECCエラー信号752をメモリ制御器70及び75に送信する。比較器634及び636の場合と同様に、ECC比較器638は、入力信号が安定したときに比較が行われるようにタ

イミングどりされる。

読み取り動作の場合には、32ビットデータ及び7ビットECCがメモリアレイ600から読み取られる。更に、ECC発生器642は、ECC比較器638がメモリアレイ600からの7ビットECCと比較するところの32ビットデータから7ビットECCを発生する。又、ECC比較器638は、入力信号が安定したときに比較が行われるようにタイミングどりされる。2つの信号が等しくない場合には、エラー処理及び制御論理回路625がエラーを指示し、ECCエラー信号752をメモリ制御器70及び75へ送信する。

前記したように、エラー処理論理回路625は、読み取り動作中に生じる第1ECCエラーの一次及びミラーアドレスをセーブする。論理回路625内のカウンタのECCエラーカウンタは、これが最初に発生する場合に1にセットされる。その後ECC読み取りエラーが生じると、メモリモジュール60内のECCエラーカウンタが増加される。CPU40及び50は、メモリモジ

ュールに記憶されたアドレス及びカウント情報を通常の診断テストの一部分として周期的にポーリングする。ポーリングプロセスの一部分によりこれらレジスタがクリアされ、ECCエラーのある次のアドレスをトラップできるようにする。CPU40及び50が修正されたデータをトラップされたアドレスに書き込むときには、メモリアレイ600からのこれら「ソフトエラー」が修正される。

3. メモリ制御器

前記したように、メモリモジュール60は、メモリへのデータ信号の比較は行わない。一次及びミラーメモリ制御器70及び75がこのような比較を行う。メモリ制御器70及び75は、各々、メモリモジュール60へのCPU40及び50のアクセスを制御する。一次メモリ制御器70が第7図に詳細に示されており、ミラーメモリ制御器75が第8図に詳細に示されている。これらメモリ制御器70及び75は第7図及び第8図では若干異なって示されているが、融通性を得るためにはこれらが同一であるのが好ましい。これらの図

面は説明を簡略化するために別々に示されている。第7図に示すように、一次制御及びアドレスラインは一次メモリ制御器70を通してメモリモジュール60へ直結されている。メモリ相互接続部80の一次制御信号は、READ及びWRITEといった必要な全てのタイミング及び内部制御信号を形成するために回路(図示せず)によって処理され、デコードされる。

データライン70は、書き込み動作中に書き込みバッファ715及び720を経てメモリ相互接続部85へ接続される。読み取り動作中には、メモリモジュール60からメモリ相互接続部85に送られるデータが読み取りバッファ725を通り、ECC発生器730及びECCチェック/修正回路735へ入力される。ECCチェック/修正回路735の出力は、読み取りバッファ740へ入力され、その出力はデータライン710へ接続される。

ECC発生器730は、メモリモジュール60に書き込まれるようにデータライン710から

受け取ったデータに対しECCを発生する。この発生器730からのECCは、書き込みバッファ745を経てメモリモジュール60へ送られる一次ECC信号である。

読み取り動作中にメモリモジュール60から受け取られた一次ECC信号は、読み取りバッファ748を経てECCチェック/修正回路735へ送られる。ECCチェック/修正回路735は、メモリ相互接続部85から受け取ったデータから発生されたECCをチェックして、エラーを検出する。又、回路735は単一ビットエラーを修正し、修正したデータを読み取りバッファ740を経て送信する。ECCチェック/修正回路735がエラーを修正できないと判断した場合には、修正不能読み取りエラー信号738をエラーラッチ750に送信し、該ラッチはこの信号を記憶する。

エラーラッチ750への他の入力、アンドゲート755から受け取ったECCエラー信号758である。アンドゲート755は、一次ECCエラー信号752(エラー処理及び制御論理回路

625からの)及びWRITE信号を入力として受け取る。ECCチェック/修正回路は読み取り動作に対しエラーの検出及び修正を行うので、アンドゲート755は、書き込み動作中にのみ一次ECCエラーを指示するように確保する。

エラーラッチ750への別の入力は、エラー処理及び制御論理回路625からの一次アドレス/制御エラー信号762である。エラーラッチ750への残りの入力はミラー比較不一致信号768である。このミラー不一致信号768は、一次メモリ制御器70及びミラーメモリ制御器75からメモリモジュール60へ送られた信号が互いに不一致であることを比較器が検出したときにミラーメモリ制御器75から受け取られる。

エラーラッチ750に記憶される信号は各々データビットとして記憶される。これらのビットはオアゲート760へ入力され、該ゲートは、エラーラッチ750のいずれかのビットがイネーブルされた場合に一次ミラーエラー信号をイネーブルする。

書き込み動作中には、メモリ相互接続部85のデータは、一次メモリ制御器70がメモリモジュール60へ送るものと同じデータである。このデータは、書き込みバッファ721を経て比較器765へ受け取られる。書き込み動作中に、一次メモリ制御器70からのデータがミラーメモリ制御器75からのデータに等しくない場合には、比較器765がミラー比較不一致信号768をイネーブルし、この信号は、一次メモリ制御器70のエラーラッチ750及びミラーメモリ制御器75のエラーラッチ751の両方に入力される。

読み取り動作中に、メモリモジュール60からのデータは相互接続部85から読み取りバッファ726を経て受け取られ、次いで、ECCチェック/修正回路736へ入力される。メモリモジュール60から受け取ったミラーECC信号は、読み取りバッファ749を経て受け取られると共に、ECCチェック/修正回路736にも入力される。一次メモリ制御器70のECCチェック/修正回路735と同様に、ECCチェック/修正

ミラーメモリ制御器75が第8図に詳細に示されている。ミラーアドレス及び制御信号82は、それに対応する一次制御信号80が一次メモリ制御器70を通過したのと同様にミラーメモリ制御器75に送られてデコードされる。データライン711は書き込みバッファ716を通して受け取られ、比較器765へ入力される。これらのデータラインは、又、ECC発生器731へも入力され、該発生器はミラーECC信号を発生する。ミラーECC信号は、書き込みバッファ746によってメモリモジュール60に送られる。

データライン711は、制御器75が一次制御器として働く必要がある場合に、書き込みバッファ722を経てメモリモジュール相互接続部85にも接続される。然し乍ら、一般に、1組のデータ信号しかメモリモジュール60に送られず、バッファ722は通常ディスエィブルされる。

データは、読み取り及び書き込みの両方の動作中にメモリモジュール60からメモリ相互接続部85を経てメモリ制御器75へ受け取られる。

回路736は、読み取りバッファ741を経てデータライン711にデータを出力する前に全ての単一ビットエラーを修正する。ECCチェック/修正回路736がエラーを修正できない場合には、修正不能読み取りエラー信号739をイネーブルし、この信号は一次メモリ制御器70のラッチ750の場合と同様にエラーラッチ751に記憶される。

又、エラーラッチ751は、アンドゲート758からのECCエラー信号759も記憶し、このアンドゲートは、メモリモジュール60からのミラーECCエラー信号753とWRITE信号とを合成する。更に、エラーラッチは、メモリエラーモジュール60のエラー処理論理及び制御回路625からのミラーアドレス/制御エラー信号763も記憶する。エラーラッチ751の出力はオアゲート761に入力される。オアゲート761は、エラーラッチ751のいずれかのビットがイネーブルされた場合にミラーメモリエラー信号をイネーブルする。

処理システム20'は、内部的には二重ルールシステムである。一方のルールは、CPU40と、キャッシュメモリ42と、メモリ制御器70と、内部バス46とを備えている。他方のルールは、CPU50と、キャッシュメモリ52と、メモリ制御器75と、内部バス56とを備えている。然し乍ら、メモリモジュール60は共有リソースである。従って、メモリモジュール70及び75は、メモリモジュール60に対しては二重ルール-単一ルールインターフェイスをなす。従って、本明細書のB節で述べた本明細書の原理によれば、このインターフェイスにおいてエラーチェックが与えられる。好ましい実施例においては、このようなエラーチェックが2つの異なった技術を含む。先ず第1に、CPU50からメモリ制御器75へ送られるデータ信号はメモリモジュール60に書き込まれず、CPU40からメモリ制御器70を経て送られるデータ信号と比較される。メモリ制御器75はこの比較を行うと共に、メモリモジュール60へ送られたデータに対してエラーチェッ

クを行う。メモリモジュール60は、メモリ制御器70及び75からのアドレス、制御信号及びECCを比較し、不一致を検出する。第2のエラーチェック技術は、メモリ制御器70及び75がメモリデータからそれ自身のECCを発生することを含む。

本発明の別の特徴は、単一ビットメモリエラーによってシステム欠陥を生じさせるのではなくてメモリ制御器70及び75により単一ビットメモリエラーを修正することである。この技術は、例えば、アルファ粒子が衝突することにより通常生じる単一ビットメモリエラーを受け入れる。このようなエラーを修正すると、システム欠陥時間が減少され、単一の共有のメモリモジュールを使用できるようにする。エラーの発生及び位置に注目することにより後で診断を行うことができる。例えば、所定数以上のこのような修正可能なエラーを受けるメモリボードを交換することが所望される。

メモリモジュール60と、一次及びメモリ制御器70及び75との間のインターフェイスが第7図及び第8図の左側部分に一般的に示されている。第9図は、メモリ制御器70と内部バス46及びクロスリンク90とのインターフェイス回路770を示している。メモリ制御器75にも同じインターフェイス回路が含まれる。

メモリモジュール60と、一次及びメモリ制御器70及び75との間のインターフェイスが第7図及び第8図の左側部分に一般的に示されている。

メモリ制御器70と内部バス46及びクロスリンク90とのインターフェイス回路770を示している。メモリ制御器75にも同じインターフェイス回路が含まれる。

メモリ制御器70と内部バス46及びクロスリンク90とのインターフェイス回路770を示している。メモリ制御器75にも同じインターフェイス回路が含まれる。

メモリ制御器70と内部バス46及びクロスリンク90とのインターフェイス回路770を示している。メモリ制御器75にも同じインターフェイス回路が含まれる。

メモリ制御器70と内部バス46及びクロスリンク90とのインターフェイス回路770を示している。メモリ制御器75にも同じインターフェイス回路が含まれる。

メモリ制御器70と内部バス46及びクロスリンク90とのインターフェイス回路770を示している。メモリ制御器75にも同じインターフェイス回路が含まれる。

メモリ制御器70と内部バス46及びクロスリンク90とのインターフェイス回路770を示している。メモリ制御器75にも同じインターフェイス回路が含まれる。

／＼装置からデータを読み取る場合にはバッファ 790 からの入力を、内部バス 40 のデータとして選択する。

ドライバ 782 は、メモリ制御器 70 へのデータ経路を与え、CPU がメモリモジュール 60 へ書き込みするかメモリモジュール 60 へ DMA 書き込みするか或いはメモリ再同期（スレーブ）動作をするために作動される。メモリ再同期動作については以下で詳細に述べる。これらの動作は、メモリモジュール 80 及び 80' の内容が互いに等しくセットされるようにするために使用される。メモリ再同期動作においては、データを受け取っているモジュールが「スレーブ」となり、そしてデータを送信しているモジュールが「マスター」となる。ドライバ 782 へ入力を与えるマルチプレクサ 794 は、実行されている動作が CPU のメモリ書き込みである場合にはバッファ 786 からの入力を、或いは動作が DMA 書き込みであるかメモリ再同期（スレーブ）動作である場合にはバッファ 790 からの入力をメモリモジュール 80

0 のためのデータとして選択する。

バッファ 784 はクロスリンク 90 へのデータ経路を与え、I/O 装置へ書き込みするかメモリの DMA 読み取りを行うか又はメモリ再同期（マスター）動作を行うように作動される。バッファ 784 へ入力を与えるマルチプレクサ 796 は、動作がメモリの DMA 読み取りである場合にはバッファ 788 からの入力を、或いは動作が I/O 装置への CPU 書き込みであるか又はメモリ再同期（マスター）動作である場合にはバッファ 786 からの入力をクロスリンク 90 のためのデータとして選択する。

マルチプレクサ 798 及び 799 は、アドレス及び制御信号を各々メモリ相互接続部 80 へ供給する。両方のマルチプレクサ 798 及び 799 は、DMA 又はメモリ再同期（スレーブ）動作を伴わないメモリ動作に対してはバッファ 786 の出力を、DMA 動作に対しては DMA エンジン 775 の出力を、或いはメモリ再同期（スレーブ）動作に対してはバッファ 790 の出力を、上記ア

ドレス及び制御信号のソースとして選択する。

4. クロスリンク

メモリ再同期、DMA 及び I/O 動作のためのデータはクロスリンク 90 及び 95 に送られる。一般に、クロスリンク 90 及び 95 は、CPU モジュール 30 と、CPU モジュール 30' と、I/O モジュール 100、110、120 と、I/O モジュール 100'、110'、120' との間の通信を行う。クロスリンク 90 及び 95 は同じものであるから、クロスリンク 90 の要素及び動作についてのみ説明する。

クロスリンク 90 は、第 10 図及び第 11 図に各々示された並列レジスタ及び直列レジスタを含んでいる。これら両形式のレジスタは、本発明の好ましい実施例ではプロセッサ間通信に使用される。通常の動作中には、プロセッサ 20 と 20' が同期され、クロスリンク 90/95 及び 90'/95' の並列レジスタによって各々処理システム 20 と 20' との間でデータが並列に交換される。処理システム 20 及び 20' が同期されない

ときには（ブートストラップ中に最も注目すべきことである）、データが直列レジスタによって交換される。

第 10 図に示された並列レジスタは、クロスリンク診断通信レジスタ 901、セーブ状態レジスタ 903、通信レジスタ 906 及び状態レジスタ 909 を備えている。並列レジスタのアドレスは I/O スペース内にある。この I/O スペース内においてこれらのアドレスはシステムアドレススペース内にあるか又はゾーンアドレススペース内にある。「システムアドレススペース」という用語は、システム 10 全体にわたってアクセスでき、ひいては、両プロセッサ 20 及び 20' によってアクセスできるアドレスを意味する。「ゾーンアドレススペース」という用語は、特定のクロスリンクを含むゾーンのみによってアクセスできるアドレスを意味する。

クロスリンク DCR（診断通信レジスタ）901 は、クロスリンク 90 内に存在し、情報を交換すると共にゾーン 11' からゾーン 11 内の診

断機能を作作用させることのできる能力を有している。クロスリンクDCR901は、次のように構成された32ビットレジスタであるのが好ましい。

ビット	意味
31	受信フラグ
30:24	【指定済み: 全て0】
23:16	受信データ
15	送信フラグ
14:8	【指定済み: 全て0】
7:0	送信データ

クロスリンクは、両方のゾーンへの割込みを生じさせることができると共に、他のゾーンによってデータを受け取ったり当該ゾーンでデータを利用したりできるようにする。DCRは、I/Oモジュールがロックステップ状態にあって且つクロスリンクがマスター/スレーブモードにあるときしか使用できない。一方のゾーン、即ちマスターゾーンは、他方のゾーン、即ちスレーブゾーンのI/O装置を制御し、スレーブゾーンのクロスリンクは通信に使用される。マスターゾーンのク

定する。その並列状態レジスタ909は、クロスリンクがスレーブモードにセットされたときにはI/O装置と共にゾーン11に与えられているので、使用することができない。

明らかなように、クロスリンクDCRのアドレスは、読み取り及び書き込みに対しシステム及びゾーンアドレススペースの両方になければならない。特に、各ゾーンは、それ自身のクロスリンクDCR（ゾーンアドレススペース）及び他のゾーンのクロスリンクDCR（システムアドレススペース）から個々に読み取ることができねばならない。同様に、各ゾーンは、それ自身及び他のゾーンのDCRに書き込みできねばならない。

セーブ状態レジスタ903は、処理システム20を再ブートするための状態情報を含んでいる。CPU40及び50は、このような情報をシステムアドレススペースにおいてセーブ状態レジスタ903に書き込む。というのは、同期動作中にデータがレジスタ903に記憶されるからである。同期動作へと再ブートするためには、同じ状態情

ロスリンクDCRは使用されない、発生される割込みは、スレーブゾーンのCPU及びマスターゾーンのCPUへ至る。各ゾーンは、データの送信及び受信に対して割込みを使用するかポーリングを使用するかについてそれ自身の選択を行う。

ゾーン11が全てのI/O装置を制御する場合には、次のようなシーケンスを経てゾーン11'に通知する。

1. ゾーン11'のクロスリンクDCRにビット23:16を書き込む。
2. 割込みをイネーブルする（ゾーン11'の並列状態レジスタ909）
3. 割込みのための待機
4. 送信器の空き割込みは、ゾーン11'がゾーン11'のクロスリンクDCRを読み取ったことを報告する。
5. 受信器いっぱい割込みは、ゾーン11'が応答することを報告する。

ゾーン11'は、その並列状態レジスタを用いて、それ自身の割込みを以下に述べるように設

報をクロスリンク90及び95とクロスリンク90'及び95'とに書き込みが必要である。ブートストラップは非同期な動作であるので、各CPU40及び50は、それ自身のクロスリンクしかアクセスしない。従って、状態レジスタ903はゾーンアドレススペースから読み取られる。

通信レジスタ906は、ゾーン間で交換されるべき独特のデータを含んでいる。このようなデータは、通常は、各ゾーンにとって独特のメモリソフトウェアカウンタのような非同期データである。レジスタ906のデータが独特であるから、書き込みのための通信レジスタ906のアドレスは、ゾーンアドレススペース内にある。アドレスがシステムスペースにある場合には、情報が両方のゾーンにおいて通信レジスタへ書き込まれ、もはや独特ではなくなる。

然し乍ら、読み取りのための通信レジスタ906のアドレスはシステムアドレススペース内にある。従って、同期動作の間には、両方のゾーンが一方のゾーンから通信レジスタを読み取れると

同時に他方のゾーンから通信レジスタを読み取る
ことができる。

状態レジスタ909は、クロスリンク90に
対して多数の状態ビットを含んでいる。状態レジ
スタ909は、好ましくは32ビット長さであり、
次のようなビット指定を有している。

ビット	値	意味
31:14	0	【指定済み】
13	1	割込みイネーブル(ビット10-12)を セット
12	1	DCRバッファ空き割込みをイネーブル
11	1	DCR受信割込みをイネーブル
10	1	クロック位相エラー割込みをイネーブル
9	1	DCRバッファ空きフラグ。DCRが書き 込まれるときには0にセットそしてDCR が読み取られるときには1にセット。
8	1	DCR受信いっぱいフラグ。全てのDCR が書き込まれるときには1にセットそして DCRが読み取られるときには0にセット。
7:6	X	クロック位相エラービットはクロック位相 エラーの際にセットされそして直列状態レ ジスタを通してリセットされる
5	0	
4:3	00	通常のパス動作(リードオンリ)
	01	メモリ再同期スレーブ
	10	メモリ再同期マスター
	11	二重同期要求
2	0	
1:0	00	クロスリンクオフ(リードオンリ)
	01	クロスリンクオンスレーブ

マスター」は、モジュール相互接続部間の同期は
とるが、CPU間の同期はとらない。

「クロスリンクオン二重」は、両CPU40
及び40'がモジュール相互接続部130及び1
30'を制御できるようにすると共に、CPU5
0及び50'が相互接続部132及び132'を
制御できるようにする。「クロスリンクオン二重」
は、CPUの同期をとると共にI/O相互接続部
の同期をとる。これらCPUを同期させるには、
両ゾーンの全てのメモリレジスタ及びバスサイ
クルが同一であることが必要である。通信レジ
スタ906を経て独特のデータを取り扱わねばなら
ない。モジュール相互接続部の同期とは、クロス
リンク90、96、90'及び95'が同じ又は同
等のバスサイクルでそれらの各々のモジュール相
互接続部130、132、130'及び132'
を駆動することを意味する。

クロスリンク90が「マスターモード」にあ
るときには、CPU40及び50が4つ全部のモ
ジュール相互接続部130、132、130'及

10 クロスリンクオンマスター

11 クロスリンクオン二重

メモリ再同期動作及びクロック位相エラーは、

以下で詳細に説明する。ビット0及び1によつて
識別されるスレーブ、マスター及び二重モードは、
処理システム20と20'との間のクロスリンク
通信の形式を示している。

「クロスリンクオフ」とは、並列クロスリン
ク910で20と20'との間の通信が行えない
ときの状態を意味する。このモードは、ゾーン間
の同期をとらない。「クロスリンクオンスレーブ」
とは、或るゾーンがそのモジュールを他のゾーン
に相互接続するために使用する状態を意味する。
ゾーンAが「クロスリンクオンスレーブ」である
場合には、ゾーンBが「クロスリンクオンマスタ
ー」となる。従つて、CPU40'及び50'は、
モジュールの相互接続部130及び132と13
0'及び132'に対して制御権を有する。C
PU40及び50は、相互接続部130、132、
130'及び132'へアクセスしない。「クロ
スリンクオンスレーブ」及び「クロスリンクオン

び132'を制御する。CPU40及び50がク
ロスリンク90'及び95'の状態をチェックす
るために、状態レジスタ909'の読み取り及び
書き込みアドレスがシステムアドレススペース内
にある。

セーブ状態レジスタ909、通信レジスタ9
06及び状態レジスタ909は、全て、これらが
単一の並列レジスタバス910を経て情報を転送
できるようにするインターフェイス回路を含んで
いる。並列レジスタバス910は、クロスリンク
90の他部分が並列レジスタと通信できるように
する。

直列レジスタが第11図に示されている。全
ての直列クロスリンクレジスタは、非同期通信に
使用されるので、ゾーンの特定のスペース内にあ
る。直列クロスリンクレジスタ及び直列クロスリ
ンクの目的は、プロセッサ20及び20'がロッ
クステップ同期状態(即ち位相固定クロック及び
同じメモリ状態)で動作していなくてもこれらプ
ロセッサが通信できるようにすることである。

直列クロスリンクは、6個のレジスタと1つの制御器と925を備えている。これらのレジスタには、制御及び状態レジスタ912、直列クロスリンク要求レジスタ914、直列クロスリンク応答レジスタ916、直列クロスリンク質問レジスタ918、直列クロスリンク送信レジスタ920及び直列クロスリンク受信レジスタ922が含まれる。送信レジスタ920は直列クロスリンク送信ライン921に接続され、受信レジスタ922は直列クロスリンク受信ライン923に接続される。制御器925はこれら直列レジスタの動作を整合する。

制御及び状態レジスタ912は、制御及び状態フラグと並列クロスリンクレジスタに記憶された全ての情報のコピーとを含んでいる。制御及び状態レジスタ912の内容は、次の通りである。

11	1	ル (読み取り/書き込み)
		クロック位相エラー割込みのイネーブル (読み取り/書き込み)
10	1	送信完了割込みのイネーブル (読み取り/書き込み)
09	1	応答受信割込みのイネーブル (読み取り/書き込み)
08	1	質問受信割込みのイネーブル (読み取り/書き込み)
07:06	00	無機能
	01	ループバック要求
	10	状態読み取り要求
	11	CPU質問要求の送信
05	1	再同期状態のセット (04:03の作動) (書き込み)
04:03	00	通常のバス動作 (読み取り/書き込み)
	01	メモリ再同期スレーブ
	10	メモリ再同期マスター
	11	二重同期要求
02	1	クロスリンク状態のセット (01:00の作動) (書き込み)
01:00	00	クロスリンクオフ (読み取り/書き込み)
	01	クロスリンクオンスレーブ
	10	クロスリンクオンマスター
	11	クロスリンクオン二重

ビット	値	意味
31:27	0	指定済み
26	X	ゾーンID (読み取りのみ)
25	X	クロック位相検出器エラー (読み取り/書き込み) (書き込み1はクリア)
24	1	クロスリンクDCRバッファ空き (読み取りのみ) DCRの書き込みによりクリア
23	1	クロスリンクDCR受信いっぱい (読み取りのみ) DCRの読み取りによってクリア
22	1	直列送信器ビジー (いずれかのゾーンによって要求されたアクティビティ) (読み取りのみ)
21:20	00	他のゾーンに対する自動応答 (読み取りのみ)
	01	このゾーンからのループバック要求の送信
	10	このゾーンからの状態読み取り要求の送信
	11	このゾーンからの質問の送信
19	1	送信バッファ空き
18	1	直列応答オーバーフロー (読み取りのみ) クロスリンク応答レジスタの読み取りによりリセット
17	1	直列質問いっぱい (読み取りのみ) クロスリンク質問レジスタの読み取りによりリセット
16	1	直列質問オーバーフロー (読み取りのみ) クロスリンク質問レジスタの読み取りによりリセット
15	1	直列質問いっぱい (読み取りのみ) クロスリンク質問レジスタの読み取りによりリセット
14	1	割込みイネーブルのセット (書き込み)
13	1	クロスリンクDCRバッファ空き割込みのイネーブル (読み取り/書き込み)
12	1	クロスリンクDCR受信割込みのイネーブル

レジスタ912及び他の直列レジスタの機能は、直列クロスリンクの動作の説明から理解することができよう。これら動作の1つはループバックである。ループバック動作はプロセッサ20'を含むが、このプロセッサを整合せずに行われる。直列の制御及び状態レジスタ912にループバック要求フラグをセットすると、直列要求レジスタ914の内容がゾーン11'のクロスリンク90' (即ち、90'又は95')へ送られる。クロスリンク90'はメッセージを直列応答レジスタ916へ返送する。このループバック機能により、クロスリンクケーブルを照合できると共に、一方のゾーンが他方のゾーンが電力を有することを確認することができる。

もう1つの動作は状態の読み取りであり、これは、一方のゾーンのクロスリンクが他方のゾーンのクロスリンクの状態を読み取れるようにする。直列の制御及び状態レジスタ912に状態読み取り要求フラグをセットすると、要求レジスタ914が状態情報の要求をクロスリンク90'に送信

するようにされる。このメッセージを受け取ると、クロスリンク90'はその直列の制御及び状態レジスタ912'の内容をクロスリンク90の直列応答レジスタ916に送信する。

第3の機能は、汎用のクロスリンクデータ転送であるデータ転送機能である。転送されるべきデータ（好ましい実施例では32ビット）を直列クロスリンク要求レジスタ914へ書き込んだ後に、直列の制御及び状態レジスタ912にCPU質問要求フラグをセットすると、要求レジスタ914の内容がクロスリンク90'に送られ、該クロスリンクはこれらの内容を質問レジスタ918'に保持する。自動応答は発生されないが、CPU40'にデータは得られる。CPU40'による応答は、クロスリンク90の直列質問レジスタ918へ送られる。

与えられる最後の機能は、直列割込み及び状態である。好ましい実施例では、直列のクロスリンクは、次の事象に基づいて割込みを発生することができる。

I/Oを発生する。制御器925は、ドライバ、バッファ、マルチプレクサ及び遅延要素を含んでいる。遅延要素は、同期のために追加される。コンピュータシステム10は高速動作であると共に密接な同期が要求されるので、適切な動作及び同期を維持するためにはケーブルの延びのような固有の信号遅延がクロスリンクによって補償されねばならない。例えば、以下で詳細に述べるように、I/Oからの読み取り中には、各CPUモジュールが別々の処理システム20及び20'のI/Oモジュールからデータを受け取る。別々のシステムからのデータは別々の経路をたどるので、このような動作により同期の欠陥が生じる。遅延要素は、同期を保持するために信号遅延及び経路の差を補償する。第12図に示すように、クロスリンク90の遅延要素は、マルチプレクサへの入力の速度を下げるのに用いられる。但し、これはマルチプレクサへの他の入力に並列クロスリンク通路25から送られるときである。

制御器925においては、ドライバ933が

直列要求及び送信の完了；

直列応答の完了（ループバック又は状態応答に対し）；

直列質問受信の完了；

診断通信レジスタのデータ利用；

診断通信レジスタのバッファ空き；及び

ゾーン間クロック位相エラー。

割込みは、1つの共通の割込みベクトルで個々にマスクされる。

更に、ここに示す実施例では、制御及び状態レジスタ912の幾つかの状態ビットが直列リンクの状態及びマシンの再同期状態に関連している。これらのビットは次の通りである。

クロスリンク同期状態；

イネーブルされた割込み；

ベンディングの割込み；及び

受信オーバーフローインジケータ。

クロスリンク90の制御器925が第12図に示されている。制御器925の制御デコーダ930は、以下で詳細に述べる規定に従って信号A

信号Aによってイネーブルされてデータをメモリ制御器70へ送信する。ドライバ936は、信号Bによってイネーブルされて、並列レジスタバス910を経て並列レジスタへデータを送信する。ドライバ939は信号Cによってイネーブルされ、モジュールの相互接続部130へデータを送信する。ドライバ942は信号Dによってイネーブルされて並列クロスリンク通路25へデータを送信する。

制御器925のバッファは、外部で発生したデータを受信するのに用いられる。バッファ945はメモリ制御器70からのデータを受け取りそしてバッファ948はバッファ945の出力に現われるそのデータを直列レジスタへ送信する。バッファ951は並列クロスリンク通路25からデータを受け取る。バッファ957は並列レジスタから並列レジスタバス910を経てデータを受け取る。

各マルチプレクサは、特定の行き先に対して別々のデータソースを選択する。マルチプレクサ

960は信号Eによって制御され、バッファ951から並列クロスリンク通路25を経てデータを受け取ると共に、遅延要素980及びマルチプレクサ963からデータを受け取る。マルチプレクサ960の出力はドライバ933を経てメモリ制御器70へ送られる。

マルチプレクサ963は信号Fによって制御され、並列レジスタバス910からバッファ957を経て入力を受け取り、直列レジスタから入力を受け取りそしてモジュール相互接続部130からバッファ954を経て入力を受け取る。マルチプレクサ963の出力は、遅延要素980を通過した後にマルチプレクサ960へ入力される。

マルチプレクサ966は信号Gによって制御され、並列クロスリンク通路25からバッファ951を経て入力を受け取ると共に、メモリ制御器70からバッファ945及び遅延要素985を経て入力を受け取る。マルチプレクサ966の出力はドライバ936を経て並列レジスタバス910へ送られる。

統されているか)ミラーレールにあるか(即ち、ミラーメモリ制御器75に接続されているか)によって決まる。

一般に、次のような10個の機能を実行することが必要である。

他のゾーンを読み取る；

他のゾーンを書き込む；

このゾーンの直列レジスタを読み取る；

このゾーンの直列レジスタを書き込む；

並列レジスタのゾーンアドレスを読み取る；

並列レジスタのゾーンアドレスを書き込む；

並列レジスタのシステムアドレスを読み取る；

並列レジスタのシステムアドレスを書き込む；

I/Oを読み取る；そして

I/Oを書き込む。

第13A図ないし第13P図は、実行されるべき種々の機能と、クロスリンクの種々のモード及び状態とに対する信号AないしIの状態を示している。各々の状態及び機能を説明するのではなく、第13A図ないし第13P図をいかに用いる

マルチプレクサ969は信号Hによって制御され、並列クロスリンク通路25のデータをバッファ951によって受け取ると共に、メモリ制御器70のデータをバッファ945及び遅延要素985によって受け取る。マルチプレクサ969の出力はドライバ939を経て相互接続部130に送られる。

マルチプレクサ970は信号Iによって制御され、並列レジスタからバス910及びバッファ957を経て入力を受け取り、メモリ制御器70からバッファ945を経て入力を受け取りそしてモジュール相互接続部130からバッファ954を経て入力を受け取る。マルチプレクサ970の出力は、ドライバ942によって並列クロスリンク通路25へ送られる。

信号A-D及び選択コードE-Iの状態は、実行されるべき動作、クロスリンクモード(即ち、二重、マスター又はスレーブ)及びクロスリンクの状態(オン/オフ)と、クロスリンクが一次レールにあるか(即ち、一次メモリ制御器70に接

かを理解するために2つについてのみ詳細に説明する。

第13A図は、二重モードにあるときのクロスリンク90に対する(即ち、ゾーン11の一次レールにおける)制御信号の状態を示している。I/O読み取り動作においては、制御信号A及びDが「オン」であり、B及びCが「オフ」である。信号Aはドライバ933をイネーブルしてデータがメモリ制御器70へ通過するように確保し、そして信号Dはドライバ942をイネーブルしてデータが並列クロスリンク通路25を経てクロスリンク90へ通過するように確保する。メモリ制御器70及び並列クロスリンク通路25へデータを送る際に含まれるマルチプレクサは、信号E、F及びIによって各々制御されるマルチプレクサ960、963及び970である。制御信号Eは、マルチプレクサ963の出力に対応する入力1を選択するようにセットされる。制御信号Fは、マルチプレクサ963がモジュール相互接続部130からのデータに対応する入力3を選択するよう

にセットされる。制御信号Iは、マルチプレクサ970がモジュール相互接続部130を選択するようにセットされる。

従って、この信号選択状態では、モジュール相互接続部130からのデータがマルチプレクサ963及び960とドライバ933とを経てメモリ制御器70へ送られる。これは、I/O読み取りトランザクションのためのデータ経路である。又、このデータは、マルチプレクサ970及びドライバ942を経て並列クロスリンク通路25へ送られ、これが適当であるのは、クロスリンク90が二重モードにあるためである。

第13E図は、同じ状態に対する制御信号を示しているが、メモリ制御器90はマスターモードにあり、従って、クロスリンク90'に信号を送信しない。この状態のI/O読み取り動作は、モジュール相互接続部130からのデータがメモリ制御器70へ送られるように確保するために同じ信号設定を含む。信号Dが「オフ」であり、Iに対する選択信号がないので、並列クロスリンク

ードとして知られている冗長形態で同じタスクを実行するときには、CPUモジュール30及び30'が同じ速度で動作を実行することが重要である。さもなくば、処理システム間を再同期させてI/Oとプロセッサ間とのエラーチェックを行うのに多量の処理時間がかかることになる。処理システム20及び20'の好ましい実施例においては、基本的なクロック信号が同期され、互いに位相固定される。耐欠陥コンピュータシステム10は、処理システム20及び20'へのクロック信号の周波数を制御すると共に各処理システムのクロック信号間の位相差を最小にするためのタイミングシステムを備えている。

第14図は、処理システム20及び20'において実施される本発明のタイミングシステムのブロック図である。このタイミングシステムは、処理システム20のCPUモジュール30に含まれた発振システム200と、処理システム20'のCPUモジュール30'に含まれた発振システム200'とを備えている。発振器200'の要

通路25を経てメモリ制御器90'へ至るデータ通路はなく、これは、クロスリンク90がマスターモードにあるときの動作として選んでいる。

メモリ制御器90が二重モードにある状態でのI/O書き込み動作の場合には、「オン」になる唯一のドライバ制御信号が信号Cであり、これはドライバ939をイネーブルしてモジュール相互接続部130へデータを通せるようにする。それ故、制御する必要のある唯一のマルチプレクサはマルチプレクサ969であり、第13A図は、制御信号Hがメモリ制御器70からのデータを選択することを示している。第13図のクロスリンク95（二重モードにあるゾーン11のミラーレール）の動作は、第13B図のクロスリンク90に対する制御信号と若干異なる。これは、以下で述べるように、I/O書き込み動作中には、各I/Oモジュールへのデータが両プロセッサ20及び20'から送られるからである。

5. 発振器

両方の処理システム20及び20'が二重モ

ードは、発振器200の要素と同じであり、両方の発振システムの動作は同じである。従って、発振器200と200'の動作が異なる場合を除いて、発振器200の要素及び動作のみについて説明する。

第14図に示すように、発振システム200の大部分、特にデジタル論理回路は、クロスリンク95の内部に存在するが、その配置は本発明にとって重要ではない。発振システム200は、電圧制御式のクリスタル発振子(VCXO)205を備えており、これは、好ましくは86.88MHzの基本的な発振信号を発生する。VCXO205の周波数は、その入力電圧レベルによって調整することができる。

クロック分配チップ210は、基本的な発振信号を分割し、好ましくは、全て同じ周波数を有する4つの主たるクロックを発生する。主CPU40については、クロックがPCLK_L及びPCLK_Hであり、これらは論理的に互いに逆のものである。ミラーCPU50については、ク

ロック分配チップ210がクロック信号MCLK L及びMCLK Hを発生し、これらも論理的に互いに逆のものである。これらクロック信号のタイミング及び位相関係は、第15図に示されている。好ましくは、クロック信号PCLK L、PCLK H、MCLK L及びMCLK Hの周波数は約33.33MHzである。又、クロックチップ210は、第15図に示すように16.16MHzの位相固定ループ信号CLKC Hも発生する。この位相固定ループ信号はクロック論理回路220に送られ、該回路はこの信号をバッファする。

クロック論理バッファ220は、CLKC H信号を同期とりを用いるために発振器200'に送信する。発振器200'のクロック論理バッファ220'は、それ自身のバッファされた位相固定ループ信号CLKC' Hを発振器200の位相検出器230に送信する。又、位相検出器230は、クロック論理バッファ220から遅延要素225を経て、バッファされた位相固定ループ

信号CLKC Hを受信する。遅延要素225は、クロック論理バッファ220'からのケーブルの延長による遅延を近似する。

位相検出器230は、その入力の位相固定ループ信号を比較し、2つの出力を発生する。その1つは、位相差信号235であり、これはループ増幅器240を経てVCO205の電圧入力に送られる。この位相差により、増幅器240は、位相差を補償するようにVCO205の周波数を変更する信号を発生する。

位相検出器230の他の出力は、あり得べき同期欠陥を指示する位相エラー信号236である。

第16図は、位相検出器230の詳細な図である。位相検出器230は、位相比較器230と、アナログ電圧比較器234とを含んでいる。位相比較器232は、遅延素子225からのクロック信号(CLKC H)及び発振器200'からの位相固定ループクロック信号(CLKC' H)を受け取り、これら信号の位相差をアナログ電圧レベル235を発生する。

処理システム20がクロック同期の目的で「スレープ」となった場合には、スイッチ245が「スレープ」位置（即ち、閉位置）となり、アナログ電圧レベル235は、ループ増幅器240によって増幅された後に、VCO205の周波数を制御する。両スイッチ245及び245'が「マスター」位置にある場合には、処理システム20及び20'が位相固定されず、非同期で（独立して）動作する。

アナログ電圧レベル235は、位相の進み及び遅れの許容範囲を渡す2つの基準電圧Vref1及びVref2と共に電圧比較器234へ入力される。位相差が許容範囲内である場合には、PHASE ERROR（位相エラー）信号が作用されない。位相差が許容範囲から外れ、電圧比較器234がVref1ないしVref2の電圧範囲から外れる位相差信号235を感知した場合には、位相エラー信号が作用され、クロックデコーダ220を経てクロスリンク95へ送られる。

6. I/Oモジュール

第17図は、I/Oモジュール100の好ましい実施例を示している。I/Oモジュール100及び100'は同一のものであるから、モジュール100についてのみ説明する。更に、I/Oモジュール110及び120（従って、モジュール110'及び120'）はモジュール100と同様であるが、異なるI/O装置に接続されているために全く同じ形態をとることはない。

I/Oモジュール100は、二重レールモジュール相互接続部130及び132によってCPUモジュール30に接続されている。各々のモジュール相互接続部は、各々ファイアウオール1000及び1010によって受け入れられる。ファイアウオール1000及び1010は、モジュール相互接続部130及び132を経て送信されるデータの一致性をチェックするチェックバス1005によって相互接続される。このチェックは、CPUモジュール30及び30'のロックステップ同期によって行われ、これは、CPUモジュール30及び30'からI/Oモジュール100へ

書き込まれたデータをファイアウォール1000及び1010に同時に得られるようにする。

第18図は、ファイアウォール1000の要素及び好ましい実施例を示している。ファイアウォール1000は、モジュール相互接続部130への32ビットバスインターフェイス1810と、第17図に示されたバス1020へ接続するための32ビットバスインターフェイス1820とを備えている。これらのインターフェイス1810及び1820は、内部ファイアウォールバス1815によって相互接続され、該バスはファイアウォール1000の他の要素にも接続されている。バス1815は、16ビット巾又は32ビット巾の並列バスであるのが好ましい。

又、ファイアウォール1000は、CPUモジュール30とオペレータコンソールとの間でコンソールリンク1090に接続されたコンソールサポートレジスタ1830も備えているのが好ましい。コンソールリンクは、モジュール相互接続部、クロスリンク及びメモリ制御器のような通常

132からのデータは別のゾーンからのものである。ファイアウォール比較回路1840内のチェック回路がこれらデータ間の不一致を検出した場合には、割込み制御回路1880が作動されて、エラー信号を発生し、これにより、クロスリンク90及び95はCPUモジュール30及び30'に欠陥の検出を通知する。

ファイアウォール比較回路1840は、CPUモジュール30及び30'から受け取ったデータのみをチェックする。CPUモジュール30及び30'に送られるデータは、共通の発生源からのものであり、従って、チェックを必要としない。むしろ、I/O装置から受け取ったデータで、CPUモジュール30及び30'に送られるべきデータは、CRC発生器1850によって行われる繰返し冗長チェック(CRC)のようなエラー検出コード(EDC)によってチェックされる。又、CRC発生器50は、内部ファイアウォールバス1815にも接続される。

CRC発生器1850は、I/O装置によ

のデータ経路をバイパスし、コンピュータシステム10の他の部分に欠陥が生じた場合にもオペレータはCPUと通信することができる。コンソールサポートレジスタ1830は、一時的な記憶レジスタ1832、受信器1834及び送信器1836を備えているのが好ましい。レジスタ1832、受信器1834及び送信器1836の相互接続は、当業者に良く知られた一般的なものである。

ファイアウォール1000は、複数のチェック回路を含んだファイアウォール比較回路1840も備えている。ファイアウォール比較回路1840は、ファイアウォール1010内の同等の要素に接続される。ファイアウォール比較回路1840内のチェック回路は、モジュール相互接続部130から受け取ったデータをモジュール相互接続部132から受け取ったデータと比較する。一般に、両方のモジュール相互接続部がCPUモジュール30から受け入れられるとしても、モジュール相互接続部130からのデータは一方のゾーンからのものでありそしてモジュール相互接続部

で使用されるものと同じCRCコードを発生し、チェックする。I/Oモジュール100は、2つのEDCを発生するのが好ましい。その一方は、CRCでもあって、モジュール100が接続されたエサネットバケットネットワークのようなネットワークに対するインターフェイスとして用いられる。その他方は、第17図のディスクインターフェイス1072のようなディスクインターフェイスとして用いられる。

モジュール相互接続部は二重であるから、CPUモジュール30とI/Oモジュール100との間にはCRCの機能が必要とされない。例えば、CPUモジュール30においては、クロスリンク90がモジュール相互接続部130を経てファイアウォール1000と通信し、そしてクロスリンク95がモジュール相互接続部132を経てファイアウォール1010と通信する。

エサネットネットワーク1082から受け取ったメッセージは、第17図に示されたネットワーク制御器1080により有効CRCに対して

チェックされる。CRCとで完成するデータは、第17図に示されたローカルRAM1060に書き込まれる。ローカルRAM1060の全てのデータは、DMAを用いてメモリモジュール60へ転送される。DMA制御器1890は転送の整合をとり、転送されているCRCエンコードデータの有効性をチェックするようにCRC発生器1850に指示する。

I/O装置との殆どのデータ転送は、DMAで行われる。データは主メモリとI/Oバッファメモリとの間で移動される。転送されるデータの各論理ブロックは、CRCによってカバーされる。エサネットの場合には、論理ブロックがメッセージパケットである。メッセージは、DMAにより、主メモリ60からファイアウォール1000を通してローカルRAM1060へ転送される。ファイアウォール1000のCRC発生器1850は、メッセージに対してCRCを計算する。

DMA制御器1890は、データ転送に対するCRC発生器1850の動作を制御する。論理

通信媒体と少なくとも同程度の信頼性を有するものであることが好ましい。種々のI/Oモジュール、例えば、同期プロトコルを取り扱うモジュールは、適当なプロトコルのCRCコードを発生してチェックするCRC発生器を有しているのが好ましい。

一般に、DMA制御器1890は、共有のメモリ制御器1050及びアドレスされているローカルRAM1060に対して特定のDMA動作の一部分を処理する。32ビットバス1020は、2つの異なったモードで駆動される。DMA設定中には、DMA制御器1890はバス1020を標準的な非同期マイクロプロセッサバスとして使用する。DMA動作が行われるローカルRAM1060のアドレスは、共有メモリ制御器1050及びDMA制御器1890によって供給される。実際のDMA転送中には、DMA制御器1890がバス1020を同期状態で駆動するようにDMA制御ライン1895に指示する。共有メモリ制御器1050は、バス1020の各バスサイクル

ブロックが転送されたときには、DMA制御器1890がCRC発生器1850からの発生されたCRCを読み取り、それを、ローカルRAM1060に記憶されたデータに加える。ネットワーク制御器1080は、ローカルRAM1060からエサネットネットワーク1082へデータを転送すると、CRCをチェックする。CRCコード自体を除く全てのエサネットパケットは、メモリモジュール60へ転送される。CRCのエラーはCRC発生器1850によって識別され、割込み制御器1880を介して報告される。

ディスクサブシステムとのデータのやり取りは、エサネットインターフェイスと同様に行われる。CRC発生器1850は、ディスク制御器1072によって使用される特定のCRCコードを発生し又はチェックする。これは、I/Oモジュール100の場合と同様に単一レールシステムに存在するか又はこれを通して転送されているデータがエラー検出コードによってカバーされるように確保し、これは、データが最終的に通過する

ごとに32ビットデータワードを転送し、DMA制御器1890は、転送されるべきワードがどれほど残されているかの経過を保持する。又、共有メモリ制御器1050は、ローカルRAM1060を制御し、次のDAMアドレスを発生する。

I/Oモジュール(100、110、120)は、それら自身のローカルRAM1060に対する読み取り/書き込み動作を制御する役目を果たす。CPUモジュール30は、メモリアレイ60との転送動作を制御する役目を果たす。メモリ制御器70及び75(第9図に示す)のDMAエンジン775は、CPUモジュール30に対しDMA動作を指示する。このような労力の分割によりいずれかのモジュールのDMA論理の欠陥がゾーン11又は11'の値のモジュールにおけるデータ完全性を低下しないようにする。

又、ファイアウォール1000は、I/Oモジュール100に対し他の重要な機能も実行する。ファイアウォール1000のI/O診断制御レジスタ1860は、クロスリンク診断制御レジスタ

901と同じ構造を有し、CPU40、50、40'及び50'と診断マイクロプロセッサ1100との間で通信を行うことができる。診断マイクロプロセッサ1100のこの間接的な接続により、該マイクロプロセッサがコンピュータシステム10の他のモジュールに影響を及ぼすことが防止される。

トレースRAM1872及びトレースRAM制御器1870の機能については以下で詳細に述べる。簡単に述べると、欠陥が検出されてCPU及びCPUモジュール30及び30'に通知されたときには、コンピュータシステム10全体にわたる種々のトレースRAMが以下に述べる幾つかの機能を実行するようにされる。トレースRAMとの通信は、トレースバス1095を経て行われる。トレースRAM制御器1870は、トレースバス1095からの信号に回答して、トレースRAM1872が記憶を停止するか又はその内容をトレースバス1095にダンプするようにさせる。

好ましくは32ビットの並列バスであるI/O

モジュールバス1020は、ファイアウォール1000及び1010と、I/Oモジュール100の他の要素とに接続される。共有メモリ制御器1050も、I/Oモジュール100のI/Oバス1020に接続されている。共有メモリ制御器1050は、32ビット+パリティを有する共有メモリバス1065によってローカルメモリ1060に接続されている。好ましくは、ローカルメモリ1060は、メモリ容量が128KBのRAMであるが、RAM1060のサイズは任意である。共有メモリ制御器1050及びローカルRAM1060は、I/Oモジュール100に対するメモリ容量を貸している。

ディスク制御器1070は、第1図のディスク1075、1075'のようなディスクに対する標準的なインターフェイスを与える。ディスク1075は、SCSI（小型コンピュータ標準インターフェイス）バスの実施のごとき標準バスインターフェイス1072によってディスク制御器1070に接続されるのが好ましい。又、ディス

ク制御器1070は、ローカルRAM1060を使用するか又はI/Oモジュールバス1020と通信するために共有メモリ制御器1050に接続される。

ネットワーク制御器1080は、ネットワークインターフェイス1082により、エサネットネットワークのような標準ネットワークにインターフェイスする。又、ネットワーク制御器1080は、ローカルRAM1060及びI/Oモジュールバス1020の両方に対するインターフェイスとして働く共有メモリ制御器1050にも接続される。然し乍ら、I/Oモジュールバス1020のいずれか1つの特定の構成又は構造にする必要はない。

システムサポート及びコンソール要素1040は、I/Oモジュールバス1020にも接続される。システムサポート及びコンソール要素1040は、ASCIIインターフェイス1042を介してオペレータコンソールのインターフェイスを与える。オペレータコンソールは、年のような或る情報を入力できるだけでなく、診断の目的に

監視できるようにすると共に、冷却システムがそれらの適切な動作を確保できるようにする。PCIMサポート要素1030は、或る欠陥又は欠陥のおそれの指示。例えば、バッテリー電圧が許容できないほど低下したという指示があったときだけメッセージを受け取るのが好ましい。又、PCIMサポート要素1030を用いて全ての電力及び冷却サブシステムを周期的に監視することもできる。PCIMサポート要素1030及びI/Oモジュール100'の同等の要素は、プロセッサ20及び20'のサポートシステムが機能不良になった場合に耐欠陥コンピュータシステム10が或るゾーンを停止できるようにする。

システムサポート及びコンソール要素1040は、I/Oモジュールバス1020にも接続される。システムサポート及びコンソール要素1040は、ASCIIインターフェイス1042を介してオペレータコンソールのインターフェイスを与える。オペレータコンソールは、年のような或る情報を入力できるだけでなく、診断の目的に

も使用することができる。オペレータコンソールは、コンソールリンク1090及び1091を経てファイアウオール1000及び1010のコンソールサポートレジスタ1830を経てCPUとデータを交換する。

又、診断マイクロプロセッサ1100は、I/Oモジュールバス1020にも接続されている。診断マイクロプロセッサ1100の動作については以下で詳細に述べる。一般に、マイクロプロセッサ1100は、欠陥が検出されたときにトレースRAM1872のようなトレースRAMからのエラーチェック情報を収集するのに用いられる。このデータは、トレースバス1095及び1096に集められ、ファイアウオール1000及び1010を各々通り、モジュールバス1020を経てマイクロプロセッサ1100へ送り込まれる。

C. プロセッサ間及びモジュール間通信

1. データ経路

コンピュータシステム10の要素はそれ自体で耐欠陥システムを構成しない。正常の動作中に

り欠陥の発生を他のゾーンに指示する1本のラインとを備えているのが好ましい。

耐欠陥コンピュータシステム10は、ほとんどの過渡欠陥が生じて二重レールシステムとして連続的に動作するように設計されている。I/Oサブシステム(モジュール100、110、120、100'、110'及び120')も、過渡欠陥を受けることがあるが、動作を続けることができる。好ましい実施例においては、ファイアウオール比較回路1840によってエラーが検出されると、CPUで指示された動作に対しライン25を介して同期マシンチェックを行うことが要求される。CPU30及び30'のソフトウェアは、欠陥の生じた動作を再トライする。DMAによって指示された動作の場合には、同じエラー検出によってライン25を経て同期割込みが生じ、CPU40、50、40'及び50'のソフトウェアがDMA動作を再スタートさせる。

幾つかの過渡エラーは直ちに修復されない。例えば、CPUモジュール30のエラー信号76

通信を行うことができそして欠陥検出及び修正中に動作を行うことのできる通信経路及びプロトコルが必要とされる。このような通信にとって重要なことはクロスリンク通路25である。クロスリンク通路25は、並列リンク、直列リンク及び既に述べたクロック信号で構成される。これらは、第19図に示されている。並列リンクは、2つの同じ組のデータ及びアドレスライン(16)、制御ライン(7)、割込みライン(7)及びエラーライン(7)を含んでいる。16本のデータ及びアドレスラインと、7本の制御ラインは、CPUモジュール間で、例えば、モジュール制御接続部130及び132(又は130'及び132')から或いはメモリモジュール60(60')から交換されるべき情報を含んでいる。

7本の割込みラインは、4つの異なるI/O割込みレベルを表わす4本のラインと、両ゾーンに対するコンソール「ホルト」要求を同期させる1本のラインと、両ゾーンに対するマシンチェックを同期させる1本のラインと、CPU/メモ

2によって指示される制御エラーは、メモリモジュール60に未知のデータを生じさせる。この状態においては、CPUモジュール30は、もはやフェイルセーフシステムの一部として確実な機能できないので、除去される。従って、メモリアレイ60は、CPU30がシステムに再び加わる前にメモリ再同期動作を受けねばならない。CPU/メモリ欠陥ラインは、CPU30に欠陥が生じたことをCPU30'に指示する。

サイクルの状態、バイトマスク、方向及びレディ状態の組み合わせを表わしている7本の制御ラインは、CPUモジュール(30及び30')とI/Oモジュールとの間にハンドシェイクを与える。サイクルの状態は、実行されているバス動作の形式、即ち、CPUによるI/Oの読み取り、DMA転送、DMA設定又は割込みベクトル要求を指示する。「バイトマスク」は、ローカルRAM1060において32ビットワードのデータの1バイト(8ビット)を修正できるように、16本のデータラインのどれが有効なデータを含んで

いるかを指示する。「方向」は、DMA動作に対してトランシーバ経路の選択を固定する。DMA転送は所定のソースと行き先との間で生じるので、データ経路を選択するのに関連した時間オーバーヘッドを、データ転送のごとに支払う必要はない。「レディ」メッセージは、要求された動作の完了を指示するためにCPUとI/Oモジュールとの間に送られる。

任意の時間にいずれかの処理システム20又は20'を作動して制御ラインの状態を読み取ることができる。作動したり停止したりすることは処理システム20と20'との間で同期して行われる。

直列クロスリンクは、状態読み取り、ループバック、データ転送、状態の転送割込みに対し、直列データ転送を行うために2組の2本のラインを備えている。

交換されるクロック信号は、位相固定クロック信号CLKC H及びCLKC' H(遅延)である。

リンク90へ送られる。このときには、データがメモリ制御器70及びクロスリンク90'まで転送される。第12図に示すように、クロスリンク90は、ファイアウォール1000からメモリ制御器70へのデータを遅延し、クロスリンク90'へのデータが「捕獲」に十分な時間を有し且つ処理システム20及び20'が同期状態に保たれるようにする。次いで、データは、クロスリンク90からメモリ制御器70を通りそして内部バス46を経てCPU40へ送られる。同様に、クロスリンク90'へのデータもCPU40'へ送られる。

データをCPU50及び50'へ読み込むために同様の経路がとられる。共有メモリ制御器1050からのデータは、ファイアウォール1010を経てクロスリンク95へ送られる。このときには、データがクロスリンク95'へ送られると共に、クロスリンク95内の遅延ユニットに通される。

I/O装置からデータを読み取るときには、

第20A図ないし第20D図は、CPUモジュール30及び30'とI/Oモジュール100及び100'の要素(これらを介して種々の作動中にデータが通過する)のブロック図である。これら要素の各々については既に述べた。

一般に、動作中のデータ経路は対称的である。例えば、CPUのI/O読み取りは、互いにミラー像関係にある経路を経てI/Oモジュール100及びI/Oモジュール100'に対して行われる。このような場合に、図面は全てのデータ経路を示しているが、その1つのみについて詳細に説明する。

第20A図は、CPUのI/O読み取り動作用のデータ通路を示している。例えば、ディスク1075(1075')又はネットワークからのデータは、共有メモリ制御器1050(1050')を通して転送するためにローカルRAM1060(1060')に記憶されるものとする。1つの経路については、データがモジュール相互接続部130のファイアウォール1000を経てクロス

1つの読み取り動作しか必要とされない。「ミラー」CPUのI/O読み取り動作は、処理システム20'のI/O装置から共有メモリ制御器1050'を経て受け取ったデータ及びI/O装置100'のローカルRAMから受け取ったデータに対して行われる。

I/Oモジュール100、110及び120は、各々、I/Oモジュール100'、110'及び120'と同様であってそれに対応しているが、対応するI/Oモジュールは互いにロックステップ同期状態にない。CPUのI/O読み取りに対してメモリ制御器1050'及びローカルRAM1060'を使用して、データは先ずクロスリンク90'及び95'へ送られる。他のデータ経路は、メモリ制御器1050からの経路と同等である。データは、クロスリンク90'及び95'からメモリ制御器70'及び75'を経て最終的にCPU40'及び50'へ各々送られる。同時に、データは、各々、クロスリンク90及び95へ送られ、次いで、遅延要素を通ることなく、各

々、CPU40及び50まで送られ続ける。

第20B図は、CPUのI/O書き込み動作を示している。データは、2つの別々のCPU I/O書き込み動作において、CPU40、50、40'及び50'を経て共有メモリ制御器1050及び1050'とローカルRAM1060及び1060'とに転送される。CPU40によって転送されるデータは、CPUのI/O読み取り中のデータとは逆方向に同じ経路に沿って送られる。特に、このようなデータは、メモリバス46、メモリ制御器70、遅延ユニット985（同期をとれるようにするための）、ファイアウォール1000及びメモリ制御器1050を経て送られる。CPU50'からのデータも、CPU I/O読み取りの経路を逆方向にたどる。特に、このようなデータは、内部バス56'、メモリ制御器75'、クロスリンク95'、クロスリンク95、ファイアウォール1010及び共有メモリ制御器1050を経て送られる。上記したように、ファイアウォール1000及び1010は、I/O書き込み

で、クロスリンク90'及び95'に通される。クロスリンク95'からのデータは、クロスリンク95及びファイアウォール1010を通過して共有メモリ制御器1050へ送られる。

CPUのI/O書き込み動作と同様に、このデータ経路と、ファイアウォール1000'及び1010'を通過する対称的な経路は、ゾーン間エラーチェックを与える。ルール間エラーチェックは、メモリモジュール70、75、70'及び75'において行われる。

DMA書き込み動作のためのデータ経路が第21D図に示されており、これはCPUのI/O読み取りの場合と同様である。特に、共有メモリ制御器1050'からのデータは、ファイアウォール1000'、クロスリンク90'（遅延を伴う）及びメモリ制御器70'を経てメモリアレイ600'へ送られる。同時に、このデータは、ファイアウォール1010'、クロスリンク95'（遅延を伴う）及びメモリ制御器75'に通され、このときには、ルール間エラーチェック中にメモ

リ制御器70'からのデータと比較される。或いは又、CPUのI/O読み取りの場合と同様に、DMA書き込み動作のデータは、同等の動作中に共有メモリ制御器1050まで送られてもよい。

動作中にデータをチェックし、出力の前にエラーをチェックする。

CPU50及び40'からのデータは、処理システム20'のI/O装置への次のCPU I/O書き込みの順に、対称的な経路を経て共有メモリ制御器1050'へ送信される。CPU50及び40'からのデータは、ゾーン間チェックのためにファイアウォール1000及び1010によって比較される。各ゾーンに対するルール間チェックは、メモリ制御器70、75、70'及び75'において既に実行されている。

第21C図は、DMA読み取り動作のためのデータ経路を示している。メモリアレイ600からのデータは、メモリ制御器70及び75とクロスリンク90及び95とに同時に送り込まれる。クロスリンク95は、そのデータを、ファイアウォール1000及び共有メモリ制御器1050へ送信するまで遅延する。

メモリアレイ800'からのデータは、メモリ制御器70'及び75'に同時に通され、次い

り制御器70'からのデータと比較される。或いは又、CPUのI/O読み取りの場合と同様に、DMA書き込み動作のデータは、同等の動作中に共有メモリ制御器1050まで送られてもよい。

又、クロスリンク90'からのデータは、クロスリンク90及びメモリ制御器70を経てメモリアレイ600へ通される。クロスリンク95'からのデータは、クロスリンク95'及びメモリ制御器75'に通され、このときには、同時のルール間チェック中にメモリ制御器70'からのデータと比較される。

メモリ再同期動作のためのデータ経路が第20E図に示されている。メモリ再同期動作については第23図に関連して以下で詳細に述べる。メモリ再同期中のデータ経路を理解するために、メモリ再同期の一部分であるバルクメモリコピー機能動作が第21図に示されている。

第21図に示すように、バルクメモリコピー動作の第1の段階は、クロスリンクをメモリ再同期マスター/スレーブモードにセットして第20

E図に示すデータ経路を形成できるようにすることである(ステップ2100)。第13I図ないし第13L図は、クロスリンクがスレーブモードにあるときに通信が本質的に一方向であることを示している。従って、クロスリンクがスレーブモードにある処理システムのメモリにデータを書き込むことができるが、このメモリから情報を読み取ることはできない。更に、第13E図ないし第13H図は、第13I図ないし第13L図と共に、クロスリンクがマスターモードにある処理システムがそのメモリモジュールに書き込みを行うたびに、クロスリンクがスレーブモードにある処理システムもその同じデータを書き込む。従って、クロスリンクがスレーブモードにある処理システムは、クロスリンクがマスターモードにある処理システムと共に、そのときまでの情報を記憶するメモリモジュールを有している。

次いで、第21図のバルクメモリ転送フローチャートに示されたように、マスター処理システムは、そのメモリモジュールを検査し、そのモジ

ュールは、全てのマスターメモリの転送が完了する(ステップ2150)まで繰り返される。

全てのマスターメモリの転送が完了すると(ステップ2150)、マスタープロセッサはそのCPUレジスタの全ての内容を後で再ブートストラップ動作に使用するためにメモリに記憶する(ステップ2160)。CPUレジスタの内容がマスターメモリに記憶されることにより、それらがスレーブメモリにも記憶される。次いで、バルクメモリ転送動作が開始されて、再同期プロセスを開始することができるようにされる。

メモリ再同期動作においては、第20E図のメモリアレイ600'、即ち「マスター」からのデータがメモリ制御器70'及び75'、クロスリンク90'及び95'、クロスリンク90及び95、並びにメモリ制御器70及び75を同時に通過し、このときに、レール間エラーチェックのためにデータが比較される。メモリ制御器70からのデータは、メモリアレイ600に入れられる。

2. ユールドブートストラップ

ユールの全てのメモリページを見つける(ステップ2110)。これら全てのページが既知になると、マスター処理システムのDMAエンジンの待ち行列に入れられる(ステップ2120)。好ましくは、マスター処理システムのDMAエンジンは、DMA要求の待ち行列を含み、各要求は、転送の形式の指示、スタートアドレス、転送されるべき要素の数及び行き先の指示を含むのが好ましい。スレーブ処理システムへの全てのDMA転送の行き先は、スレーブ処理システムのメモリモジュールである。

次いで、メモリ再同期動作により、スレーブ処理システムのDMAエンジンへコマンドを送信し(ステップ2130)そして次のページのDMA転送を開始して完了する(ステップ2140)という対話プロセスが開始される。スレーブのDMAエンジンへのコマンドは、DMA転送のサイズ及びスタートアドレスを指示し、スレーブ処理システムのメモリモジュールがマスターモジュールと同一に構成されるよう確保する。これらのス

メモリ再同期動作を除けば、前記したデータ経路は、全て、ゾーン11及び11'がロックステップ同期状態にあるときに生じる。然し乍ら、耐欠陥コンピュータシステム10は、プロセッサ20及び20'がロックステップ同期状態にないときの動作も受け入れねばならない。これら状態の1つは、初期化時の「コールド」ブートストラップ動作中に生じる。別の状態は、以下に述べるメモリ再同期動作中に生じる。

第22A図ないし第22H図は、CPUモジュール30及び30'をロックステップ同期状態にもっていくブートストラップシーケンスを示すフローチャートである。

一般に、ステップ2228のようなエラー状態においては、他のゾーンに或る刺激に対する不適当な応答が与えられる。他のゾーンからの適当な応答を受け取り損なったことに対する応答は、そのゾーンによって生じたエラーを記録することである。この情報は、現場のサービス要員によってエラーを分析するのに使用することができる。

更に、他のゾーンからの適当な応答を受け取り損なったことに対する応答は、そのゾーンによって生じたエラーを記録することである。第22A図ないし第22H図においては、ゾーンが一般にゾーンA及びBと称される。とうのは、上記した動作が各々のゾーンに適用されるからである。

コールドブートストラップ動作に入ると(ステップ2200)、クロスリンク通路がオフに切り換えられ、プロセッサが通常の動作に対してセットされる(ステップ2202)。通常の動作とは、CPUがそれらの各々のメモリ制御器を直接的に制御する状態を指す。次いで、CPUは、CPUに配置されたEEPROMからそれ自身のブートストラップトークンを読み取る(ステップ2204)と共に、他のゾーンのCPUからブートストラップトークンを読み取る(ステップ2206)。両方のプロセッサは、次いで、ゾーンAからの応答を待機し(ステップ2208)、ブートストラップ動作の目的でゾーンAが「ブートマスター」となるかどうかを判断する。もしそうであ

ゾーンAは、先ず、そのオペレーティングシステムをブートする。ブートプロセスの一部分には、システム10に存在する全てのI/O装置を制御するのに必要なテーブル記述及びルーチンをロードすることが含まれる。然し乍ら、ゾーンAのこれらI/O装置のみが初期化され、アプリケーションプログラムに利用できるようにされる。

ブートマスターは、ブートシンプレックス手順に入り(ステップ2222)、これは、ゾーンAがゾーンBからのメッセージの受信を待機する(ステップ2224)状態で開始する。ゾーンBからメッセージを受信すると、ゾーンAは、ゾーンBがゾーンAと接続する要求をなしたかどうかの判断をする(ステップ2226)。もしそうでなければ、プロセスはエラー状態2228へ入り、ブートストラップ手順が完了され、ゾーンは非同期状態に保たれる。ゾーンBが接続の要求をなした場合には、ゾーンAがゾーンBに「イエス」の応答を出し(ステップ2230)そしてゾーンAが再びゾーンBの応答を待機する(ステップ22

れば、両方のシステムは、ゾーンBがブートストラップ動作のためのブートマスターとなると考えているかどうか判断する(ステップ2212)。もしそうであれば、ゾーンBは非ブート状態にセットされ、ブートマスターとならない(ステップ2214)。

ステップ2210の判断の結果としてゾーンAがブートマスターとならない場合には、両ゾーンがゾーンBがブートマスターであるかどうか判断する(ステップ2216)。ゾーンBがブートマスターでない場合には、ゾーンAがブートマスターとしてセットされる(ステップ2218)。

ステップ2210-2218が行われてブートマスターが決定された後に、各ゾーンはブートマスターであるかどうかそれ自身に尋ねる(ステップ2220)。ブートマスターとなったゾーンは、第22B図のステップ2222へ進む。他のゾーンは、第22F図のステップ2284へ進む。第22A図ないし第22H図のフローチャートの説明上、ゾーンAがブートマスターであるとする。

32)。ゾーンBがI/Oを供給しない場合には(即ち、ゾーンBがそのI/O装置の制御を保持する場合には)、別のエラー状態に入り(ステップ2236)、ブートストラップ手順が終了される。

ゾーンBがI/Oを供給する場合には、ゾーンAがゾーンBのクロスリンク状態を読み取る(ステップ2238)。ゾーンBのクロスリンク状態がゾーンBがクロスリンクスレーブであることを指示しない場合には(ステップ2240)、エラー状態に入り(ステップ2242)そしてブートストラップ手順が再び終了される。

ゾーンBのクロスリンク状態が該ゾーンがクロスリンクスレーブモードにあることを指示する場合には(ステップ2240)、ゾーンAがそれ自身のクロスリンク状態を「マスター」にセットする(ステップ2244)。ゾーンBのI/O装置が初期化され、アプリケーションプログラムに対して利用できるようにされる。ゾーンAはゾーンBからの要求を待機する(ステップ2246)。

ゾーンAが再同期要求を受け取らない場合には（ステップ2248）、エラー状態に入り（ステップ2250）そしてブートストラップ手順が完了される。

ゾーンAがゾーンBから再同期要求を受け取る場合には（ステップ2248）、ゾーンAがゾーンBに「イエス」の応答を与え（ステップ2252）そしてゾーンBからの応答を待機する（ステップ2254）。ゾーンAは、ゾーンBが再同期の準備ができるまで待機する（ステップ2256）。

ゾーンBが再同期メモリに対して構成されている場合には、そのクロスリンク状態が「クロスリンクオンスレーブ」及び「再同期スレーブ」となる。再同期スレーブモードは、ゾーンBのCPUをそれらのメモリ制御器から外すようにロックする。ここで、ゾーンBのメモリ制御器は、それに関連したクロスリンクから全ての指示を受け取る。これは、ゾーンBのメモリをゾーンAで制御するようにする。ゾーンBが再同期の準備ができ

全て、ソフトウェア及びハードウェアの実行を制御するCPUモジュールのレジスタ常駐情報である。この情報は、CPUの汎用レジスタ、アドレス変換テーブル、プロセス状態ワード、スタックポインタ、プログラムカウンタ、割込み状態、構成レジスタ及びインターバルタイマを含むが、これに限定されるものではない。これらの値は、メモリアレイ60にセーブされる。メモリ再同期がまだ有効であるから、値はメモリアレイ60'にもセーブされる。セーブされた値のメモリアドレスは、クロスリンクのセーブ状態レジスタに書き込まれる。従って、ゾーンAの全ての処理は保留され、キャッシュメモリがフラッシュされる。アプリケーション処理を再スタートするのに必要な全ての情報はメモリアレイ60及び60'に常駐する。

次いで、ゾーンAは、デュープレックス同期要求をセットし（ステップ2274）そしてゾーンBのクロスリンク状態を読み取る（ステップ2276）。ゾーンBの状態が該ゾーンがデュープ

た場合には、ゾーンAがゾーンBのクロスリンクの状態を再び読み取り（ステップ2258）そしてゾーンBの状態が該ゾーンが再同期スレーブモードにあることを指示するかどうか判断する（ステップ2260）。もしそうでなければ、別のエラー状態に入り（ステップ2262）そしてブートストラップ手順が終了される。

ゾーンBのクロスリンク状態が該ゾーンが再同期スレーブモードにあることを指示する場合には（ステップ2260）、ゾーンAがそれ自身の状態を再同期マスターモードにセットし（ステップ2264）そしてメモリ転送の待ち行列を形成する（ステップ2266）。これは、第21図について述べるバルクメモリ転送である。

次いで、ゾーンAは、DMAがメモリ再同期に対して完了するまで待機し（ステップ2268）、そのシステム状態をメモリから読み取り（ステップ2270）そして再同期の後にゾーンがスタートするメモリ位置である再スタートベクトルをセーブする（ステップ2272）。システム状態は、

レックス同期モードにないことを指示する場合には、エラー状態に入り（ステップ2280）そしてブートストラップ手順が終了する。ゾーンBがデュープレックス同期モードにある場合には、ゾーンAがデュープレックス同期要求をセットし、第22H図のステップ2308で始まる同期動作に入る。

ゾーンAの動作と同時に、ゾーンBはブートストラップのための別の組のステップを実行する。ブートストラップ技術においてゾーンBがスレーブであると決定された後に、該ゾーンはゾーンAと接続する要求を発し（ステップ2284）そして応答を待機する（ステップ2286）。ゾーンAの応答がゾーンBに接続できないという応答である場合には、ゾーンBがコンソール状態に入る。

或る状態のもとでは、1つのゾーンのみと動作することが所望される。1つのゾーンはブートできるが、他方のゾーンは拡張診断を実行する。従って、ゾーンAはゾーンBをブートしないように命令され、ゾーンBが接続の要求を出したとき

には、許可が与えられない。次いで、ゾーンBは、コンソールオペレータからの命令をコンソール状態で待機する。

ゾーンAがゾーンBの接続要求に対して「イエス」の応答を出した場合には、ゾーンBがそのI/Oを手放し（ステップ2290）、そのクロスリンク状態を「スレーブ」モードにセットしそして再同期動作を要求する（ステップ2294）。次いで、ゾーンBはゾーンAからの応答を待機する（ステップ2296）。

ゾーンBの再同期要求に対するゾーンAの応答が「イエス」でない場合には（ステップ2298）、ゾーンBがコンソール状態2300に入り、ブートストラップ動作が終了される。ゾーンAがゾーンBの再同期要求に対して「イエス」の応答を出した場合には（ステップ2298）、ゾーンBが「再同期レディ」メッセージをゾーンAに送り（ステップ2302）そしてそのクロスリンク状態を「スレーブ」モードにセットする（ステップ2304）。次いで、ゾーンBは、同期動作に

入るまでゾーンAからの再同期を待機する。

ブートストラップルーチンがまだ終了されないか又はコンソール状態に入らない場合には、両ゾーンがゾーンAのクロスリンク状態を読み取る（ステップ2308）と同時に、ゾーンBのクロスリンク状態を読み取る（ステップ2310）。両方のクロスリンク状態が等しい場合には（ステップ2312）、ブートストラップが完了しそしてブートストラップ手順が終了される。

クロスリンクA及びクロスリンクBの状態が等しくない場合には、クロスリンクがオフにされ（ステップ2314）そして両方のプロセッサがブートマスターとなるかどうか質問する（ステップ2316）。クロスリンク状態が不一致であることは、2つのゾーンの同期がうまくとれなかったことを意味する。ブートマスターゾーン（ゾーンA）は、スレーブゾーン（ゾーンB）の構成テーブルから装置を除去し、そして処理を続行する。ゾーンBは、コンソール状態において、コンソールオペレータが更に別の命令を与えるのを待機す

る。ブートマスターであるゾーンについては、ブートストラップルーチンが終了される（ステップ2316）。他のゾーンについては、コンソール状態に入り（ステップ2318）、次いで、ブートストラップルーチンが終了される。

3. メモリ再同期動作

メモリ再同期動作は、コールドブートストラップ手順の間と、以下に述べるように、ゾーンが同期を失うようにさせる過渡エラーが検出された後とに使用される。一方のゾーンは、好ましくは、他方のゾーンの動作を停止させることなく再同期しなければならない。このような再同期にとって重要なことは、コンピュータシステム10を完全に停止することなくメモリアレイ600及び600'を互いに等しくセットする技術である。これは、メモリ再同期によって行われる。

メモリ再同期のフローチャートが第23図に示されている。メモリ再同期のための一連のシーケンスの多くは、ブートストラップ動作に用いられたものと同様である。これらのシーケンスは、

簡単にいえば「ルーチン」と称され、第22図のブートストラップ動作のフローチャートに示された一連のシーケンスに相関される。

メモリがコピーされるゾーンは「マスター」であり、マスターメモリのコピーを受け取るゾーンは「スレーブ」と称する。マスターはクロスリンクマスターでもあり、スレーブはクロスリンクスレーブでもある。メモリ再同期の初期化時に、マスターのクロスリンクは「クロスリンクオン（マスター）」モードである。マスターが他のゾーンから再同期要求を受け取ると（ステップ2320）、マスターは「イエス」で応答し（ステップ2322）そしてスレーブからの応答を待機する（ステップ2324）。マスターは応答を受け取ると、メモリ再同期（マスター）ルーチンに入り（ステップ2326）、これは、第22B図及び第22C図に示されたステップ2256ないし2282より成る。次いで、変更されたブートストラップルーチンである自動ウオーム再スタート手順が開始される。この手順については以下で述

べる。

そのメモリが同期されることを要求しているゾーンで、スレーブとなるようなゾーンが再同期を所望しているときには(ステップ2330)、「再同期要求」をマスターへ送り(ステップ2332)そして応答を待機する(ステップ2334)。マスターからの応答を受け取ると、メモリ再同期(スレーブ)ルーチンを開始する。これは、第22G図に示すブートストラップルーチンのステップ2298ないし2306で構成される。このルーチンが完了すると、スレーブは自動ウォーム再起動手順を開始し(ステップ2338)、次いで、両ゾーンはゾーン再同期手順に入る。

ゾーン再同期ルーチンは、第22H図に示すブートストラップルーチンのステップ2308-2318より成る。唯一の相違は、第22H図に示されたステップ2316において、ゾーンがブートマスターであるかどうかではなくてクロスリンクマスターであるかどうかをそれぞれ自体に尋ねる。

4. ウォーム再起動動作

び全ての保留されたジョブが再起動される。

D. 欠陥検出、分離及び修理

1. トレースRAM

欠陥検出のための種々の方法は、CPUモジュール30及び30'とファイアウォール1000及び1010との説明において既に述べた。これらの方法は、メモリ制御器から受け取ったアドレス及び制御信号が同じであることをメモリモジュール60及び60'によってチェックすることと、修正不能なメモリデータのエラーをチェックすることを含む。又、これらの方法は、2つの「ルール」を通過するアドレス、制御及びデータ信号が同じであることをメモリ制御器の対70/75及び70'/75'によって比較することを含む。又、ファイアウォール1000及び1010は、1/0書き込み中にゾーンからのデータを比較するチェックを有している。他の欠陥検出機構も考えられる。

処理システム20及び20'の欠陥停止機能と矛盾しないように、処理システムの欠陥検出は、

ウォーム再起動手順が第24図に示されている。この手順に入ると、最初に尋ねる質問は、クロスリンクの状態が何であるかである。それが通常動作、マスター再同期動作又はスレーブ再同期動作である場合には、ウォーム再同期動作が終了される。

クロスリンク状態が「デュープレックス同期」である場合には、CPUが一時的に停止され(ステップ2420)、次いで、コンソールコードに入って(ステップ2430)、ブートストラップ動作を完了する。スレーブ状態レジスタが分析される。スレーブ状態レジスタが0である場合には、第22図に示すようにコールドブートストラップを行う必要がある(ステップ2450)。さもなければ、CPUレジスタは所定のメモリ位置からロードされ、これらはスレーブ状態レジスタの値+4であるのが好ましい(ステップ2460)。次いで、スレーブ状態レジスタによって指示された位置へ分岐がなされる(ステップ2470)。この後者の動作によりオペレーティングシステム及

欠陥が制御されない状態で他のモジュールへ伝播されないようにする。然し乍ら、フェイスセーフ機能を実現するためには、コンピュータシステム10は、欠陥の発生源を分離してシステムを修理することができねばならない。欠陥の発生源を探索するための好ましい技術は、トレースRAMと称する再循環レジスタを使用しており、これらのRAMは処理システム20及び20'のデータ経路全体にわたって配置されている。これらのトレースRAMは、それに関連したデータ経路を経て送られる連続するメッセージを記録する。欠陥が生じたときには、欠陥の発生源を探索するようにメッセージが分析される。

第25図は、コンピュータシステム10におけるトレースRAMの好ましい位置を示すCPUモジュール30及びI/Oモジュール100のブロック図である。他の位置を使用することもできる。処理システム20においては、トレースRAMが全てのデータ経路に配置されるのが好ましい。従って、第25図において、トレースRAM25

00及び2505は、各々、メモリ制御器70及び75に配置される。トレースRAM2510、2515及び2518はクロスリンク90の全てのインターフェイスに配置され、そしてトレースRAM2520、2525及び2528はクロスリンク95の全てのインターフェイスに配置される。トレースRAM1872及び1877は各々ファイアウォール1000及び1010に配置される。相補的な組のトレースRAMは、処理システム20'に配置される。

トレースRAMの選択及び配置はほぼ任意であるが、好ましくは、エラーのソースを正確に識別してエラー源を識別できるようにするためには各データ経路が少なくとも1つのトレースRAMを含んでいなければならない。

トレースRAM2800の一例が第26図に示されている。トレースRAM2600は、関連バス経路のN個の最新のサイクルに転送されたデータを記憶する円形バッファとして構成されるのが好ましい。トレースRAM2600はバッファ

レジスタ2605を備えており、その入力はそれに関連したデータ経路からデータを受け取るように接続されている。バッファ2000へのロード入力はアンドゲート2615の出力である。アンドゲート2615の入力は、データ経路からのクロック信号と、欠陥が検出されたときに発生されたFREEZE TRACE RAM信号である。バッファ2605のアドレス入力は、アンドゲート2615からカウント信号を受け取るリサイクルカウンタ2610に接続されている。

トレースRAM2600のサイズNは、最も悪いメッセージ転送に必要なバスサイクルの全数と、エラー又は欠陥が生じたときにトレースRAMに制御信号を送信するに必要とされるサイクルの全数との和によって決定される。この制御信号は、トレースRAMがそのデータを凍結してそれ以上のメッセージの記憶を停止するためのコマンドであるのが好ましい。

次いで、トレースRAMの各々は、それに関連したデータ経路におけるN個の最新のトランザ

クションのコピーをそのメモリに保持する。例えば、第25図において、トレースRAM2518は、モジュール相互接続部130におけるN個の最新のトランザクションのコピーを保持する。

2. 間欠エラー

第27図は、エラー又は欠陥が検出されたときにその発生源を分離するのに使用される手順を示すフローチャートである。エラーは、「内実」な状態を繰り返さない「間欠的」なものであるか、このような状態を繰り返すものである。或る要諦によって欠陥が検出されると、エラー信号がそれに関連したクロスリンク（即ち、ファイアウォール1000及びメモリ制御器70のクロスリンク90）に送られる。クロスリンクはCPUに割り込むか又はCPUがトラップ状態に入るようにさせる。

一般に、エラー信号は、ファイアウォール又はメモリ制御器からのものである場合には1つのサイクルでクロスリンクへ送られそしてメモリアレイからのものである場合には2つのサイクルで

クロスリンクに送られる。次いで、クロスリンクの遅延要素は、両ゾーンのCPUがエラー又は欠陥について同時に通知されるように（即ち、割込み又はトラップによって）確保する。

同時に、FREEZE TRACE RAM信号は、エラーを検出した要素からコンピュータシステム10全体にわたって伝播する（ステップ2700）。このFREEZE TRACE RAM信号は、一般に、そのゾーンのクロスリンクの到達するのに1つ又は2つのメモリサイクルを必要とし、他のゾーンのクロスリンクと、エラーが見つかったゾーンのファイアウォール及びメモリ制御器とに到達するのに別のサイクルを必要とし、そして非欠陥ゾーンのファイアウォール及びメモリ制御器に到達するのに別のクロックサイクルを必要とする。

CPUは、マシンチェックルーチンに入り（ステップ2702）そしてCPUが既にマシンチェックルーチンにある間にエラーが検出されたかどうかを判断することにより（ステップ270

5) トラップ又は割込みを処理する。エラーが検出された場合には、そのエラーが内実なエラーであるとされ、システムは、第28A図ないし第28C図に示す内実な欠陥を分離する手順を開始する。

CPUがマシンチェック手順にない場合には、検出されたエラーが間欠的であるとされる。次いで、CPUは入力フラグをセットし(ステップ2710)、そのエラーを分離する手順を開始する。この入力フラグは、割込みが生じたときにCPUがマシンチェック手順にあるかどうかを判断するためにステップ2705においてテストされるフラグである。

次いで、全てのトレースRAMデータがトレースRAMバスを介してその処理システムに関連したファイアウォールに読み込まれる(ステップ2720)。次いで、診断マイクロプロセッサは、全てのトレースRAMデータをロードし(ステップ2730)、CPU40、50、40'及び50'に転送する(ステップ2735)。

応するデータ経路におけるトレースRAMデータが等しいことを意味する場合には、他のデータ経路の存在が決定される(ステップ2754)。他のデータ経路が保持される場合には、別の経路がとられ(ステップ2756)、次の比較が行われる(ステップ2750以降)。

他の経路がない場合には、CPUは、チェックされたトレースRAM対が第1のものであるかどうか判断する(ステップ2758)。もしそうであれば、例えば、CPU読み取りの際にファイアウォール1000及び1010において最初にデータが受け取られたときにエラーが生じているはずである。この点では単一のデータレールが2つの経路に分割されているので、データは同じはずであり、従って、エラー検出器は「カルブリット(罪人)」であると考えられ、欠陥であると記録される(ステップ2760)。次いで、入力フラグがクリアされ(ステップ2762)、これはマシンチェックルーチンの終了を指示すると共に割込みをクリアし、そしてマシンチェックルーチ

次いで、各CPUは、エラー信号を送った要素でスタートし(ステップ2740)、要素の欠陥源を分離する。トレースRAMデータの比較は、2500/2505、2510/2520、2515/2525、等のような対向するレール上の同じ位置のトレースRAMからのデータを比較することによって行われる。対応するメッセージ(即ち、トレースRAMの同じ深さにあるメッセージ)が比較されて、エラー(即ち、対応するメッセージ間の不一致)が検出されたかどうか調べられる。エラー検出器においてトレースRAMが分析された後に、次のトレースRAMは、欠陥が生じた動作中にデータがたどった経路を逆にたどる。従って、CPUの読み取りに対しては、経路がCPUから離れる方向となる。

エラー検出ルーチンの最初の段階においては、CPUモジュール30及び30'の各レールからのトレースRAMデータが比較される(ステップ2750)。別々のレールにおけるトレースRAM間にエラーがなくて(ステップ2752)、対

ンが終了となる(ステップ2764)。

ステップ2754及び2758において他の経路が保持されないが現在のトレースRAM対がチェックされた最初の対ではないことが判断された場合には、エラーがその経路において手前のレベルで生じたものと判断される。というのは、これは、トレースRAMがデータの変更を発生時に記録した場合であると共にエラーが最初に生じた場合だからである(ステップ2766)。次いで、ステップ2762及び2764において、各々、入力フラグがクリアされ、そしてマシンチェックルーチンが終了される。

然し乍ら、ステップ2752においてエラーが指示された場合には、そのエラーがオペレーションソースであるかどうかについて次の判断がなされる(ステップ2770)。「オペレーションソース」とは、特定の経路に対するデータ経路の開始を短く表現したものである。もしそうでなければ、経路はオペレーションソースに向かって後退され(ステップ2775)、レール間で別の比

較がなされる（ステップ2750以降）。ステップ2770においてエラーチェックルーチンがそのときオペレーションソースであると判断された場合には、各々、そのオペレーションソースが欠陥の位置として記録され（ステップ2780）、入力フラグがクリアされ（ステップ2762）そしてマシンチェックルーチンが終了される（ステップ2764）。

3. 内実欠陥

欠陥が内実欠陥であると判断されると（第27図参照）、第28A図ないし第28C図のフローチャートに示すようにこのような内実欠陥を分離する手順が開始される。内実欠陥処理の第1のステップは、耐欠陥コンピュータシステム10をその2つの別々のゾーン1'1及び1'1'に分割することである（ステップ2800）。これは、全てのクロスリンクをオフに切り換えることによって行われる。次いで、各ゾーンの診断マイクロプロセッサがそれらゾーンに対するトレースRAMデータを読み取り（ステップ2805）そしてそ

ーを示す。分析中にハードウェア欠陥が検出された場合には、そのゾーンが除去され（ステップ2820）、報告の試みがなされ（ステップ2825）そして手順はコンソールへと続く。

トレースRAMデータのゾーン分析によりエラーがあることが示された場合には（ステップ2835）、エラーの位置が検討される（ステップ2840）。エラーがゾーン内にある場合には、欠陥がもはやハード即ち内実でないことが決定されるか又は多数の欠陥が存在することが分かり、過渡欠陥を分離しようと試みる間に生じる過渡欠陥は元の過渡欠陥を見掛け上内実であるようにする（ステップ2845）。まれなことであるが、このような状態は、エラー処理中にハードウェアの欠陥が生じた場合にたどるステップに類似した更に別の分析を必要とし、欠陥を生じたゾーンがシステムから除去され（ステップ2850）、エラーを報告する試みがなされ（ステップ2855）そして内実欠陥手順を終了してコンソールへ至る。

ステップ2840において、クロスリンクに

のデータをデュアルレールアクセスによってゾーンのCPUに送る（ステップ2810）。このプロセス中にエラーが検出された場合には（ステップ2815）、エラー処理中にハードウェア欠陥が生じており、そのゾーンがコンピュータシステム10から除去される（ステップ2820）。欠陥を報告する試みがなされる（ステップ2825）が、ハードウェア欠陥であるためにこれは不可能であり、この手順はコンソールへと続く。これにより、コンソールオペレータは詳細な診断テストを開始して、内実欠陥を探索することができる。

トレースRAMデータのデュアルレールアクセス中にエラーが検出されない場合には、そのトレースデータが分析される（ステップ2830）。この分析に用いる手順は、過渡エラー分析のステップ2740-2780に類似している。然し、ゾーンが分割されていて一方のゾーンでしかエラーが検出されないの、手順に相違が生じる。このような場合の手順は、分析を行っているゾーンではエラーを示さず、クロスリンクにおいてエラ

欠陥があると決定された場合には、欠陥が実際にはクロスリンク通路25にあるか又は他のゾーンにある（ステップ2860）。このような場合、第28B図に示すように、そのゾーンは、そのエラー分析を直列クロスリンクを経て他のゾーンへ送り（ステップ2865）そして他のゾーンが応答するかどうかについて判断がなされる（ステップ2868）。他のゾーンからの応答がない場合には（ステップ2870）、他のゾーンが不動作であると判断され、欠陥が報告される（ステップ2870）。次いで、エラー状態がクリアされ（ステップ2875）、内実欠陥分析ルーチンが完了する。

他のゾーンが応答する場合には、更に別のエラー分析が行われて、他のゾーンが欠陥状態にあるかどうか又はクロスリンクが欠陥状態にあるかどうかの判断がなされる（ステップ2877）。

ステップ2877の分析によって他のゾーンが欠陥状態にあると分かった場合には、該他のゾーンに欠陥が生じたと報告され（ステップ287

9)、エラー状態がクリアされ(ステップ2875)そして内実欠陥分析手順が完了する。

一方、ステップ2877の分析結果がクロスリンクが欠陥状態にあるというものであった場合には、直列クロスリンクを介して欠陥のあるクロスリンクを識別する試みがなされる(ステップ2881)。これは、第28A図のステップ2835と同様の手順によって行われる。欠陥のあるクロスリンクが識別されると、欠陥のあるゾーンが除去されそして欠陥が報告される(ステップ2883)。次いで、エラー状態がクリアされ(ステップ2875)そして内実欠陥の分析手順が完了する。

ステップ2835において、トレースRAMデータが分析されそして他のゾーンに欠陥があると判断された場合には、第28C図の手順をたどる。この手順において、ゾーンは直列クロスリンクを経てトレースRAMの分析結果を交換し(ステップ2885)そして他のゾーンが応答しているかどうか判断される(ステップ2887)。他

のゾーンが応答しない場合には、そのゾーンが欠陥として報告され(ステップ2889)、エラー状態がクリアされ(ステップ2891)そして内実欠陥の分析ルーチンが完了する。

応答が受信された場合には(ステップ2887)、第28B図のステップ2835のエラー分析に類似したエラー分析が実行される(ステップ2893)。エラー分析は2つの結論のうちの1つに到達する。その結論が他のゾーンが欠陥状態にあるというものである場合には、そのゾーンが欠陥として報告される(ステップ2889)。その結論がクロスリンクが欠陥状態にあるというものである場合には、クロスリンク又は他のゾーンが欠陥であると報告される(ステップ2895)。ステップ2889又は2895の報告の後に、エラー状態がクリアされ(ステップ2891)そして分析手順が終了する。

4. 修理

第27図及び第28A図ないし第28C図に示された欠陥及び内実欠陥分離手順の結果、CP

Uモジュール(即ち、CPU/メモリ/メモリ制御器の組み合わせ);クロスリンク;クロスリンク通路25;モジュール相互接続部;又はI/Oモジュール内においてエラー又は欠陥が探索されたとすることができる。一般に、間欠欠陥が検出された場合には、耐欠陥コンピュータシステム10のCPUが間欠欠陥の位置を指示し、増分カウンタが欠陥要素に対する間欠エラーの数を指示する。要素に対するカウントが所定数を越えない限りそれ以上の動作は生じない。カウントが所定数を越えて、間欠欠陥の生じる頻度が許容するには高過ぎることを指示する場合、又は検出された欠陥が内実欠陥である場合には、モジュール又は要素に欠陥があり、それらを作動不能にして、その要素を含むモジュールを交換できるまで耐欠陥コンピュータシステム10が効果的に動作を続けられるようにしなければならない。

CPUモジュールが欠陥であると分かった場合には、そのモジュールのクロスリンクを「デュープレックス」モードから「スレーブ」モードに

切り換えることによってそのモジュールを動作不能にしなければならない。「スレーブ」モードにおいては、モジュールをそのクロスリンク診断制御レジスタによって他のゾーンのCPUモジュールと通信するしかない。動作不能にされたCPUモジュールはコンソールモジュールに接続される。

クロスリンクが欠陥であると分かった場合には、欠陥状態にあるクロスリンクの部分に基づいて3つの動作のうちの1つをとることができる。メモリ制御器に接続されたクロスリンクの部分は、実質には、それに関連したCPUモジュールから区別することができない。その部分に欠陥がある場合には、CPUモジュールを動作不能にしたのと同じ手順を行わねばならない。

モジュール相互接続部に接続されたクロスリンクの部分は、実質上、モジュール相互接続部と区別することができない。この状態においてクロスリンクを動作不能にするプロセスは、モジュール相互接続部を動作不能にする手順と同じである。

クロスリンクの一部は、クロスリンク通路

と実質上区別できないクロスリンク通路25に接続される。この場合、又は欠陥がクロスリンク通路にある場合には、クロスリンクをオフに切り換えると共に、欠陥のあるクロスリンクを含むと考えられるCPUモジュールを動作不能にすることにより、ゾーンが分割される。然し乍ら、欠陥の発生源を決定できない場合には、CPUモジュール30及び30'の所定の1つが先ず動作不能にされる。エラーが持続する場合には、他のCPUモジュールが再ブートされ、予め選択されたモジュールが動作不能にされる。

同じ手順を使用して、欠陥のあるI/Oモジュールや、モジュール相互接続部や、モジュールインターフェイスに欠陥の生じたクロスリンクが動作不能にされる。3つ全ての状態において、欠陥はI/Oモジュールを伴うトランザクション中に生じ、従って、I/Oモジュールが動作不能にされる。これは、全てのCPU40、50、40'及び50'のオペレーティングシステムにそのI/Oモジュールの参照記号を除去するように通知

考え方の精神又は範囲から逸脱せずに上記細部とは異なったものが考えられる。

4. 図面の簡単な説明

第1図は、本発明による耐欠陥コンピュータシステムのブロック図。

第2図は、第1図の耐欠陥コンピュータシステムを含む物理的なハードウェアを示す図。

第3図は、第1図の耐欠陥コンピュータシステムに示されたCPUモジュールのブロック図。

第4図は、第1図に示されたコンピュータシステムの相互接続されたCPUモジュール及びI/Oモジュールのブロック図。

第5図は、第1図に示された耐欠陥コンピュータシステムのメモリモジュールを示すブロック図。

第6図は、第5図に示されたメモリモジュールの要素を示す詳細図。

第7図は、第3図に示されたCPUモジュールの一次メモリ制御器を示すブロック図。

第8図は、第3図のCPUモジュールに含ま

することによって行われる。

クロスリンク又はモジュール相互接続部の欠陥を表わす欠陥が生じた場合には、それらのモジュールを含むゾーンを動作不能にすることができる。これに対する好ましい解決策は、欠陥が生じたときに通信状態にあるI/Oモジュールを動作不能に保持することである。モジュール相互接続部及びクロスリンクの当該部分も、接続された全てのI/Oモジュールが動作不能にされたときに實際上動作不能にされる。I/OモジュールをもたないCPUモジュールは、大部分の目的に対し實際上動作不能にされる。

結論

以上、好ましい実施例について示した本発明は、前記した本発明の目的を達成する。本発明の装置及び方法について種々の修正及び変更がなされることが当業者に明らかであろう。従って、本発明は、その広い観点において、前記した特定の細部、代表的な方法及び装置、並びに解説のための例に限定されない。従って、本発明の一般的な

れたミラーメモリ制御器のブロック図。

第9図は、第7図及び第8図に示されたメモリ制御器のインターフェイス回路を示す図。

第10図は、第3図に示されたCPUモジュールのクロスリンクの並列レジスタを示す図。

第11図は、第3図に示されたCPUモジュールのクロスリンクの直列レジスタを示す図。

第12図は、第3図に示されたCPUモジュールのクロスリンクの制御器の要素を示すブロック図。

第13A図ないし第13P図は、第12図に示された制御器に対する信号の状態を示す図。

第14図は、第1図の耐欠陥コンピュータシステムのためのタイミングシステムのためのブロック図。

第15図は、第14図のタイミングシステムによって発生されたクロック信号のタイミング図。

第16図は、第14図に示されたタイミングシステムの位相検出器を詳細に示す図。

第17図は、第1図のコンピュータシステム

の I/O モジュールを示すブロック図、

第 18 図は、第 17 図に示された I/O モジュールのファイアウォール要素を詳細に示す図、

第 19 図は、第 1 図のコンピュータシステムに対するクロスリンク道路の要素を詳細に示す図、

第 20 A 図ないし第 20 E 図は、第 1 図のコンピュータシステムのデータ流れ線図、

第 21 図は、第 1 図のコンピュータシステムによるバルクメモリ転送のための論理流れ線図、

第 22 A 図ないし第 22 H 図は、第 1 図のコンピュータシステムのブートストラップ動作のための論理流れ線図、

第 23 図は、第 1 図のコンピュータシステムのメモリ再同期動作のための論理流れ線図、

第 24 図は、第 1 図のコンピュータシステムのウォーム再スタート手順のための論理流れ線図、

第 25 図は、トレース RAM の配置を含む第 1 図のコンピュータシステムのブロック図、

第 26 図は、トレース RAM のブロック図、

第 27 図は、第 1 図のコンピュータシステム

について間欠欠陥を分離するための論理流れ線図、
そして

第 28 A 図ないし第 28 C 図は、第 1 図のコンピュータシステムにおいて内実欠陥を分離するための論理流れ線図である、

図中：

10・・・耐欠陥コンピュータシステム

11、11'・・・ゾーン

12、12'・・・キャビネット

13・・・バッテリー

14・・・電力レギュレータ

16・・・冷却ファン 17・・・交流入力

20、20'・・・処理システム

30、30'・・・CPU モジュール

40、50・・・CPU

42、52・・・キャッシュメモリ

43、53・・・キャッシュバス

44、54・・・システムサポート及びキャッシュ制御器

60・・・メモリモジュール

70、75・・・一次メモリ制御器

80、82・・・バス

100、110、120・・・I/O モジュール

130、132・・・モジュール相互接続部

600・・・メモリアレイ

610・・・制御論理回路

612・・・一次制御信号デマルチプレクサ

614・・・ミラー制御信号デマルチプレクサ

618・・・一次制御信号ドライバ

618・・・システムタイミング制御信号発生器

620・・・構成エラー論理回路

625・・・エラー処理論理回路

626・・・EEPROM

630・・・比較論理回路

632・・・行及び列アドレスメモリドライバ

634・・・アドレス比較器

636・・・制御信号比較器

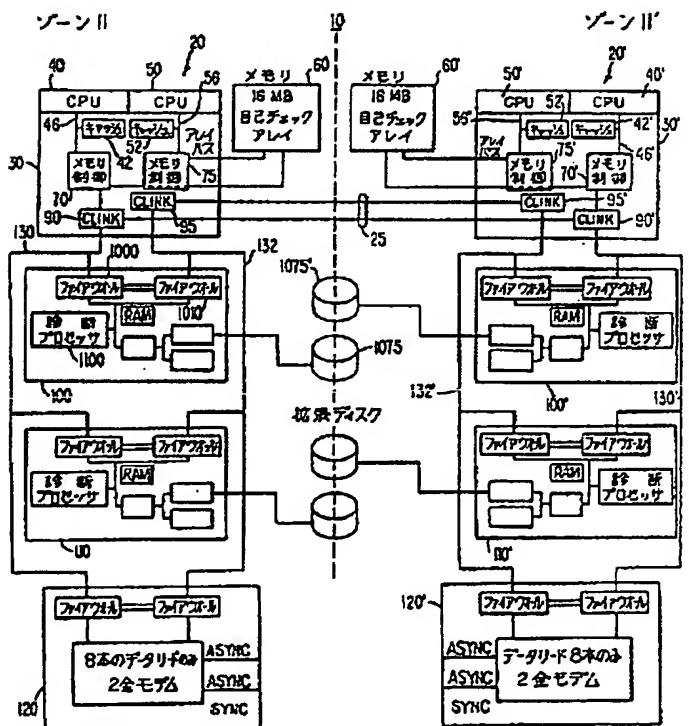
638・・・ECC 信号比較器

640・・・データ及び ECC トランシーバ

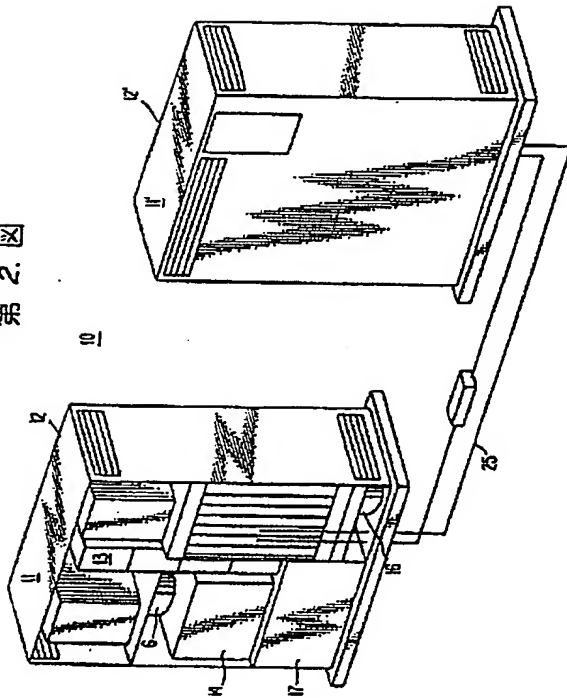
642・・・ECC 発生器

図面の浄書(内容に変更なし)

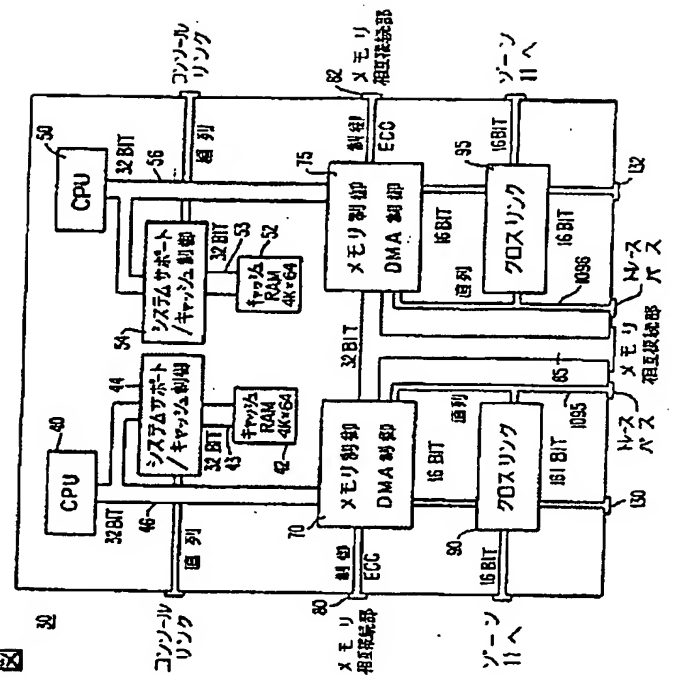
第 1 図



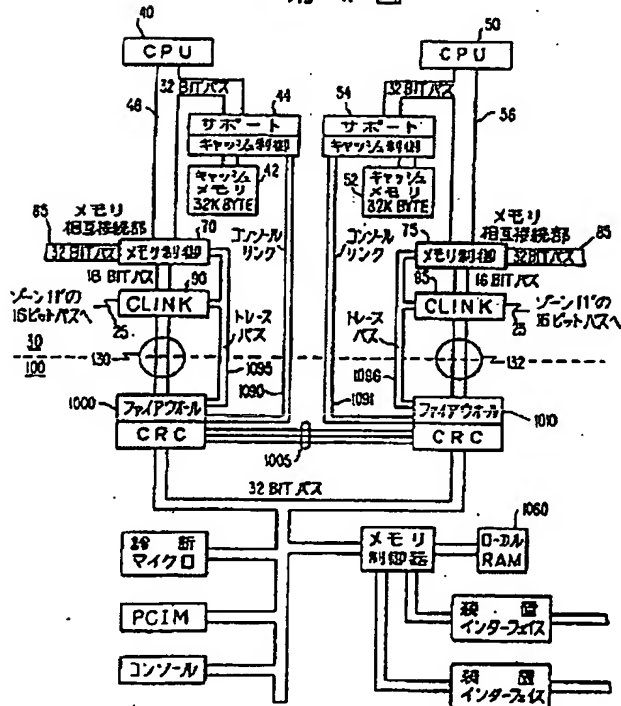
第 2 図



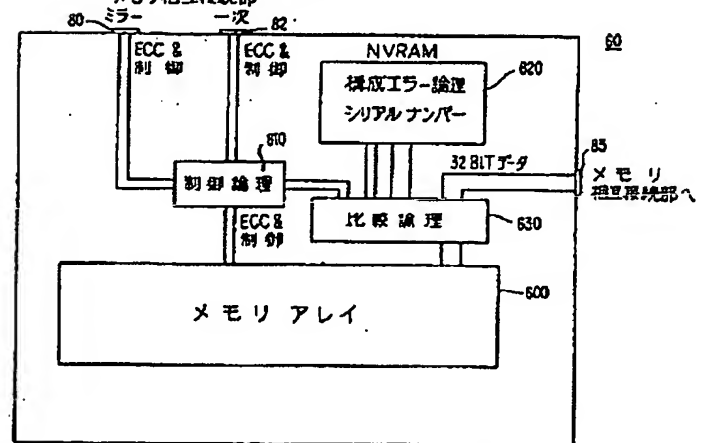
第 3 図



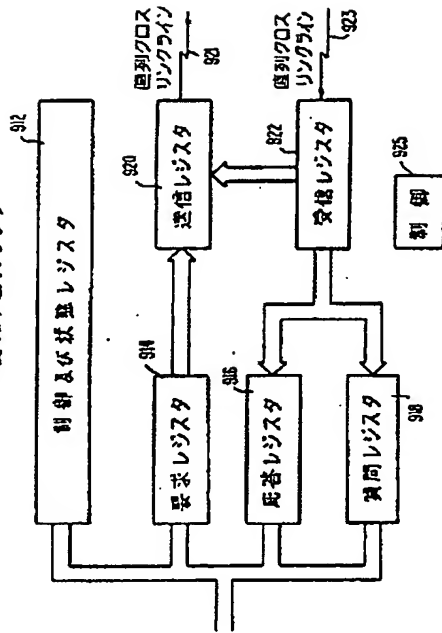
第 4 図



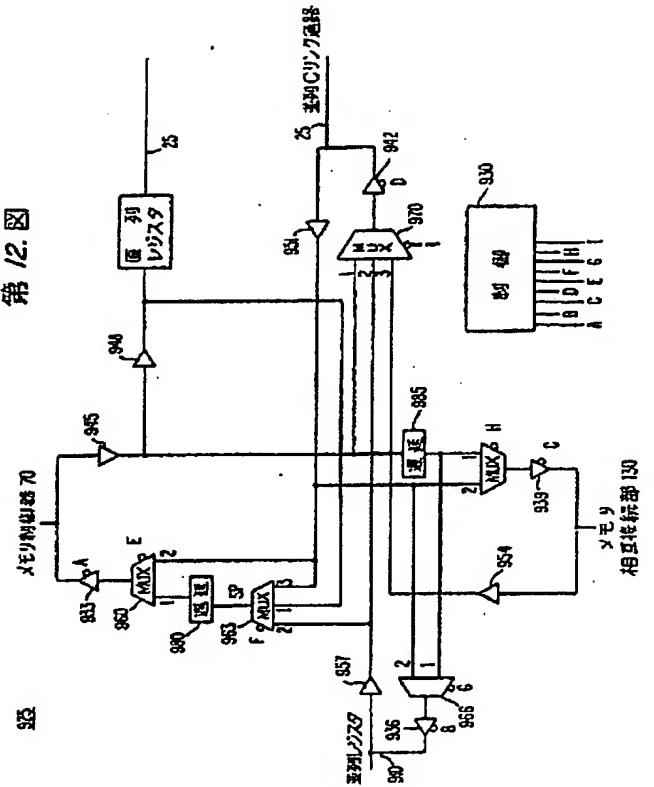
第 5 図
メモリモジュール



第11図
直列クロスリンク



第12図



第13A図

デュプレックスモードの一次レールゾーン11

制御信号	他のゾーンの読み取り	他のゾーンの書き込み	このゾーンの直列レジスタの読み取り	このゾーンの直列レジスタの書き込み	直列レジスタのゾーンアドレスの読み取り	直列レジスタのゾーンアドレスの書き込み	直列レジスタシステムアドレスの読み取り	直列レジスタシステムアドレスの書き込み	I/Oの読み取り	I/Oの書き込み
A	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF
B	OFF	OFF	OFF	OFF	OFF	ON	OFF	ON	OFF	OFF
C	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON
D	OFF	ON	OFF	OFF	OFF	OFF	ON	OFF	ON	OFF
E	2		1		1		2		3	
F										
G										
H										
I										

第 13B. 図

デュプレックスモードのミラールゾーン II

制御信号	<div> <div>他のゾーンの読み取り</div> <div>他のゾーンの書き込み</div> <div>このゾーンの直列レジスタの読み取り</div> <div>このゾーンの直列レジスタの書き込み</div> <div>並列レジスタのゾーンアドレスの読み取り</div> <div>並列レジスタのゾーンアドレスの書き込み</div> <div>並列レジスタのシステムアドレスの読み取り</div> <div>並列レジスタのシステムアドレスの書き込み</div> <div>I/O の読み取り</div> <div>I/O の書き込み</div> </div>									
	*	*	*	*	*	*	*	*	*	*
A	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF
B	OFF	OFF	OFF	OFF	OFF	ON	OFF	ON	OFF	OFF
C	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON
D	OFF	ON	OFF	OFF	OFF	OFF	ON	OFF	ON	OFF
E	2		1		1		1		1	
F			1		2		2		3	
G						1		2		
H										2
I		1					2		3	

第 13C. 図

デュプレックスモードの一次ルールゾーン II'

制御信号	<div> <div>他のゾーンの読み取り</div> <div>他のゾーンの書き込み</div> <div>このゾーンの直列レジスタの読み取り</div> <div>このゾーンの直列レジスタの書き込み</div> <div>並列レジスタのゾーンアドレスの読み取り</div> <div>並列レジスタのゾーンアドレスの書き込み</div> <div>並列レジスタのシステムアドレスの読み取り</div> <div>並列レジスタのシステムアドレスの書き込み</div> <div>I/O の読み取り</div> <div>I/O の書き込み</div> </div>									
	*	*	*	*	*	*	*	*	*	*
A	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF
B	OFF	OFF	OFF	OFF	OFF	ON	OFF	ON	OFF	OFF
C	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON
D	OFF	ON	OFF	OFF	OFF	OFF	ON	OFF	ON	OFF
E	2		1		1		1		1	
F			1		2		2		3	
G						1		2		
H										2
I		1					2		3	

第13D.図

デュプレックスモードのミラーレールゾーン11'

制御信号	<div>他のゾーンの読み取り</div> <div>他のゾーンの書き込み</div> <div>このゾーンの直列レジスタの読み取り</div> <div>このゾーンの直列レジスタの書き込み</div> <div>並列レジスタのゾーンアドレスの読み取り</div> <div>並列レジスタのゾーンアドレスの書き込み</div> <div>並列レジスタのシステムアドレスの読み取り</div> <div>並列レジスタのシステムアドレスの書き込み</div> <div>I/Oの読み取り</div> <div>I/Oの書き込み</div>									
	X	X	X	X	X	X	X	X	X	X
A	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF
B	OFF	OFF	OFF	OFF	OFF	ON	OFF	ON	OFF	OFF
C	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON
D	OFF	ON	OFF	OFF	OFF	OFF	ON	OFF	ON	OFF
E	2		1		1		1		1	
F			1		2		2		3	
G						1		1		
H										1
I		1					2		3	

第13E.図

マスターモードの一次レールゾーン11

制御信号	<div>他のゾーンの読み取り</div> <div>他のゾーンの書き込み</div> <div>このゾーンの直列レジスタの読み取り</div> <div>このゾーンの直列レジスタの書き込み</div> <div>並列レジスタのゾーンアドレスの読み取り</div> <div>並列レジスタのゾーンアドレスの書き込み</div> <div>並列レジスタのシステムアドレスの読み取り</div> <div>並列レジスタのシステムアドレスの書き込み</div> <div>I/Oの読み取り</div> <div>I/Oの書き込み</div>									
	X	X	X	X	X	X	X	X	X	X
A	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF
B	OFF	OFF	OFF	OFF	OFF	ON	OFF	ON	OFF	OFF
C	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON
D	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
E	2		1		1		1		1	
F			1		2		2		3	
G						1		1		
H										1
I		1								

第13F 図

マスターモードのミラーレールゾーン II

制御信号	<div>他のゾーンの読み取り</div> <div>他のゾーンの書き込み</div> <div>このゾーンの直列レジスタの読み取り</div> <div>このゾーンの直列レジスタの書き込み</div> <div>並列レジスタのゾーンアドレスの読み取り</div> <div>並列レジスタのゾーンアドレスの書き込み</div> <div>並列レジスタのシステムアドレスの読み取り</div> <div>並列レジスタのシステムアドレスの書き込み</div> <div>I/Oの読み取り</div> <div>I/Oの書き込み</div>									
	*	*	*	*	*	*	*	*	*	*
A	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF
B	OFF	OFF	OFF	OFF	OFF	ON	OFF	ON	OFF	OFF
C	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON
D	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
E	2		1		1		1		1	
F			1		2		2		3	
G					1		2			
H										1
I		1								

第13G 図

マスターモードの一次レールゾーン II'

制御信号	<div>他のゾーンの読み取り</div> <div>他のゾーンの書き込み</div> <div>このゾーンの直列レジスタの読み取り</div> <div>このゾーンの直列レジスタの書き込み</div> <div>並列レジスタのゾーンアドレスの読み取り</div> <div>並列レジスタのゾーンアドレスの書き込み</div> <div>並列レジスタのシステムアドレスの読み取り</div> <div>並列レジスタのシステムアドレスの書き込み</div> <div>I/Oの読み取り</div> <div>I/Oの書き込み</div>									
	*	*	*	*	*	*	*	*	*	*
A	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF
B	OFF	OFF	OFF	OFF	OFF	ON	OFF	ON	OFF	OFF
C	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON
D	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
E	2		1		1		1		1	
F			1		2		2		3	
G					1		1			
H										1
I		1								

第13H. 図

マスターモードのミラーレールゾーン 11'

制御信号	<div>他のゾーンの読み取り</div> <div>他のゾーンの書き込み</div> <div>このゾーンの直列レジスタの読み取り</div> <div>このゾーンの直列レジスタの書き込み</div> <div>並列レジスタのゾーンアドレスの読み取り</div> <div>並列レジスタのゾーンアドレスの書き込み</div> <div>並列レジスタのシステムアドレスの読み取り</div> <div>並列レジスタのシステムアドレスの書き込み</div> <div>I/Oの読み取り</div> <div>I/Oの書き込み</div>									
	*	*	*	*	*	*	*	*	*	*
A	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF
B	OFF	OFF	OFF	OFF	OFF	ON	OFF	ON	OFF	OFF
C	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON
D	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
E	2		1		1		1		1	
F			1		2		2		3	
G						1		1		
H										1
I		1								

第13I. 図

スレーブモードの一次レールゾーン 11

制御信号	<div>他のゾーンの読み取り</div> <div>他のゾーンの書き込み</div> <div>このゾーンの直列レジスタの読み取り</div> <div>このゾーンの直列レジスタの書き込み</div> <div>並列レジスタのゾーンアドレスの読み取り</div> <div>並列レジスタのゾーンアドレスの書き込み</div> <div>並列レジスタのシステムアドレスの読み取り</div> <div>並列レジスタのシステムアドレスの書き込み</div> <div>I/Oの読み取り</div> <div>I/Oの書き込み</div>									
	*	*	*	*	*	*	*	*	*	*
A	OFF	OFF	ON	OFF	ON	OFF	OFF	OFF	OFF	OFF
B	OFF	OFF	OFF	OFF	OFF	ON	OFF	ON	OFF	OFF
C	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON
D	OFF	OFF	OFF	OFF	OFF	OFF	ON	OFF	ON	OFF
E			1		1					
F		1			2					
G						1		2		
H							2			2
I							2		3	

第 13J. 図

スリープモードのミラーゾーン II

制御信号	<div>他のゾーンの読み取り</div> <div>他のゾーンの書き込み</div> <div>このゾーンの直列レジスタの読み取り</div> <div>このゾーンの直列レジスタの書き込み</div> <div>並列レジスタのゾーンアドレスの読み取り</div> <div>並列レジスタのゾーンアドレスの書き込み</div> <div>並列レジスタのシステムアドレスの読み取り</div> <div>並列レジスタのシステムアドレスの書き込み</div> <div>I/Oの読み取り</div> <div>I/Oの書き込み</div>									
	X	X	X	X	X	X	X	X	X	X
A	OFF	OFF	ON	OFF	ON	OFF	OFF	OFF	OFF	OFF
B	OFF	OFF	OFF	OFF	OFF	ON	OFF	ON	OFF	OFF
C	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON
D	OFF	OFF	OFF	OFF	OFF	OFF	ON	OFF	ON	OFF
E			1		1					
F			1		2					
G						1		2		
H										2
I							2		3	

第 13K. 図

スリープモードの一次ルールゾーン II'

制御信号	<div>他のゾーンの読み取り</div> <div>他のゾーンの書き込み</div> <div>このゾーンの直列レジスタの読み取り</div> <div>このゾーンの直列レジスタの書き込み</div> <div>並列レジスタのゾーンアドレスの読み取り</div> <div>並列レジスタのゾーンアドレスの書き込み</div> <div>並列レジスタのシステムアドレスの読み取り</div> <div>並列レジスタのシステムアドレスの書き込み</div> <div>I/Oの読み取り</div> <div>I/Oの書き込み</div>									
	X	X	X	X	X	X	X	X	X	X
A	OFF	OFF	ON	OFF	ON	OFF	OFF	OFF	OFF	OFF
B	OFF	OFF	OFF	OFF	OFF	ON	OFF	ON	OFF	OFF
C	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON
D	OFF	OFF	OFF	OFF	OFF	OFF	ON	OFF	ON	OFF
E			1		1					
F			1		2					
G						1		2		
H										2
I							2		3	

第 13L. 図

スレープモードのミラーレールゾーン II

制御信号	<div> 他のゾーンの読み取り 他のゾーンの書き込み このゾーンの並列レジスタの読み取り このゾーンの並列レジスタの書き込み 並列レジスタのゾーンアドレスの読み取り 並列レジスタのゾーンアドレスの書き込み 並列レジスタのシステムアドレスの読み取り 並列レジスタのシステムアドレスの書き込み I/Oの読み取り I/Oの書き込み </div>									
	X	X	X	X	X	X	X	X	X	X
A	OFF	OFF	ON	OFF	ON	OFF	OFF	OFF	OFF	OFF
B	OFF	OFF	OFF	OFF	OFF	ON	OFF	ON	OFF	OFF
C	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON
D	OFF	OFF	OFF	OFF	OFF	OFF	ON	OFF	ON	OFF
E			1		1					
F			1		2					
G						1		2		
H										2
I							2		3	

第 13M. 図

Cリンク オフの一次レールゾーン II

制御信号	<div> <div>他のゾーンの読み取り</div> <div>他のゾーンの書き込み</div> <div>このゾーンの並列レジスタの読み取り</div> <div>このゾーンの並列レジスタの書き込み</div> <div>並列レジスタのゾーンアドレスの読み取り</div> <div>並列レジスタのゾーンアドレスの書き込み</div> <div>並列レジスタのシステムアドレスの読み取り</div> <div>並列レジスタのシステムアドレスの書き込み</div> <div>I/Oの読み取り</div> <div>I/Oの書き込み</div> </div>									
	X	X	X	X	X	X	X	X	X	X
A	OFF	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF
B	OFF	OFF	OFF	OFF	OFF	ON	OFF	ON	OFF	OFF
C	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON
D	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
E			1		1		1		1	
F			1		2		2		3	
G						1		1		
H										1
I										

第 13N. 図

Cリングオフのミラーレールゾーン II

制御信号	<div>他のゾーンの読み取り</div> <div>他のゾーンの書き込み</div> <div>このゾーンの並列レジスタの読み取り</div> <div>このゾーンの並列レジスタの書き込み</div> <div>並列レジスタのゾーンアドレスの読み取り</div> <div>並列レジスタのゾーンアドレスの書き込み</div> <div>並列レジスタのシステムアドレスの読み取り</div> <div>並列レジスタのシステムアドレスの書き込み</div> <div>I/Oの読み取り</div> <div>I/Oの書き込み</div>									
	*	*	*	*	*	*	*	*	*	*
A	OFF	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF
B	OFF	OFF	OFF	OFF	OFF	ON	OFF	ON	OFF	OFF
C	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON
D	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
E			1		1		1		1	
F			1		2		2		3	
G						1		1		
H										1
I										

第 13O. 図

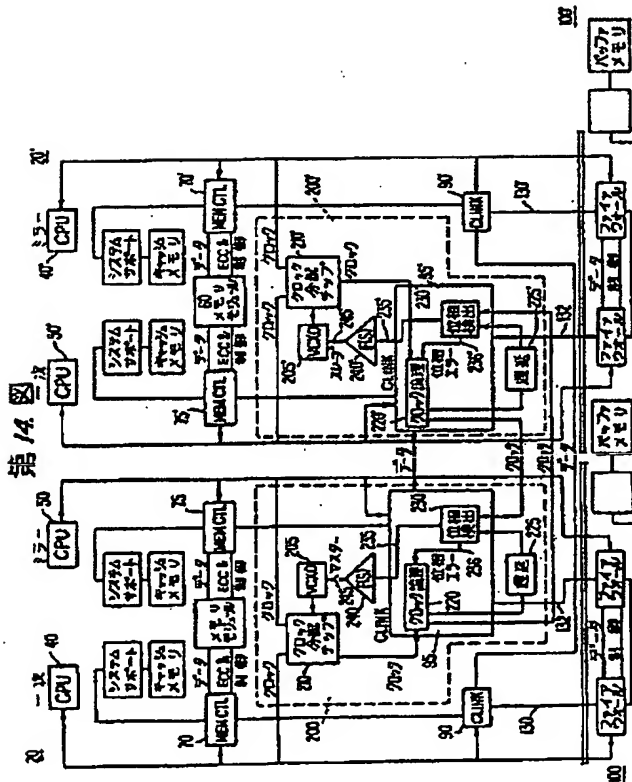
Cリングオフの一次レールゾーン II'

制御信号	<div>他のゾーンの読み取り</div> <div>他のゾーンの書き込み</div> <div>このゾーンの並列レジスタの読み取り</div> <div>このゾーンの並列レジスタの書き込み</div> <div>並列レジスタのゾーンアドレスの読み取り</div> <div>並列レジスタのゾーンアドレスの書き込み</div> <div>並列レジスタのシステムアドレスの読み取り</div> <div>並列レジスタのシステムアドレスの書き込み</div> <div>I/Oの読み取り</div> <div>I/Oの書き込み</div>									
	*	*	*	*	*	*	*	*	*	*
A	OFF	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF
B	OFF	OFF	OFF	OFF	OFF	ON	OFF	ON	OFF	OFF
C	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON
D	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
E			1		1		1		1	
F			1		2		2		3	
G						1		1		
H										1
I										

第 13 図

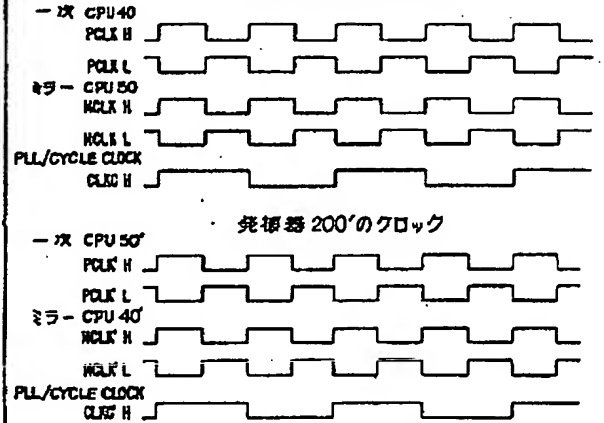
Cリングオフのミラーレールゾーン II'

		他のゾーンの読み取り	他のゾーンの書き込み	このゾーンの直列レジスタの読み取り	このゾーンの直列レジスタの書き込み	並列レジスタのゾーンアドレスの読み取り	並列レジスタのゾーンアドレスの書き込み	並列レジスタのシステムアドレスの読み取り	並列レジスタのシステムアドレスの書き込み	I/Oの読み取り	I/Oの書き込み
制御信号		X	X	X	X	X	X	X	X	X	X
A	OFF	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	OFF
B	OFF	OFF	OFF	OFF	OFF	ON	OFF	ON	OFF	OFF	OFF
C	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON	ON
D	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
E			1			1		1		1	
F					2			2		3	
G						1					
H											
I											1

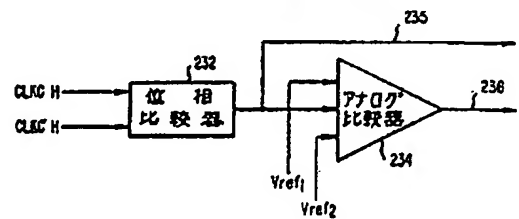


第 15 図

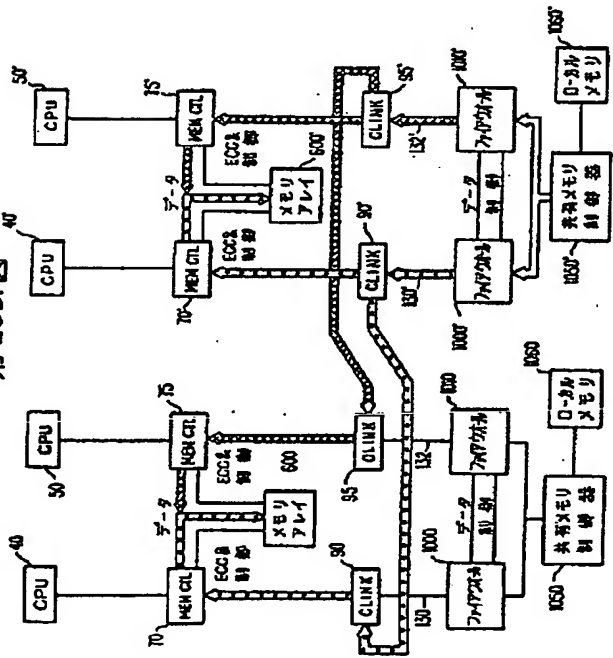
発振器 200 のクロック



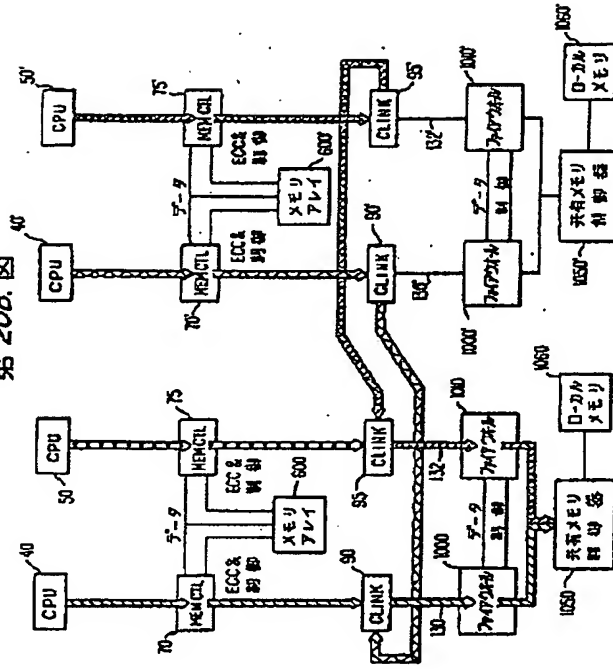
第 16 図



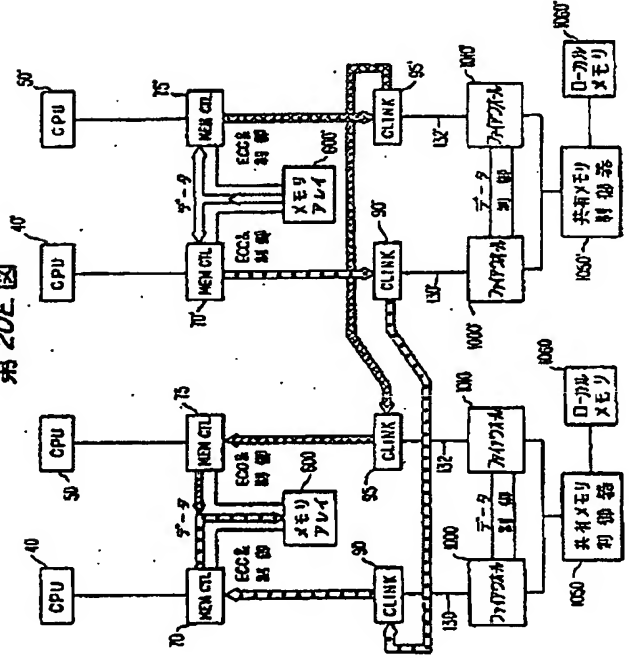
第 20D 図



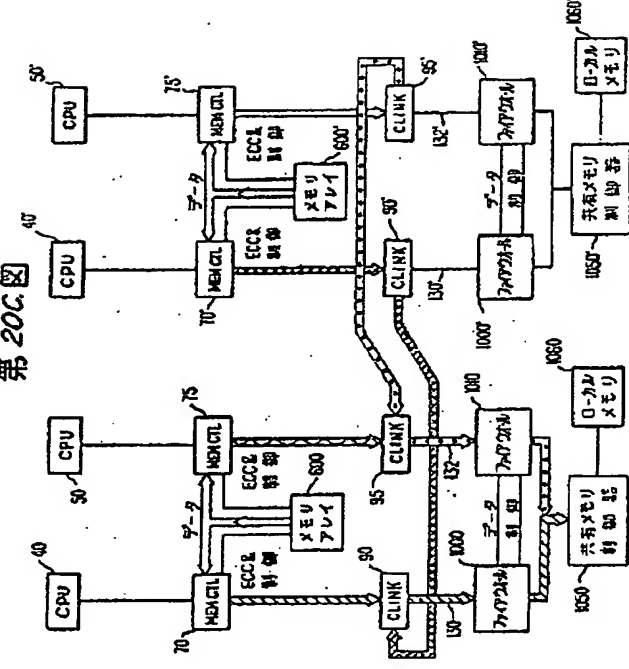
第 20B 図



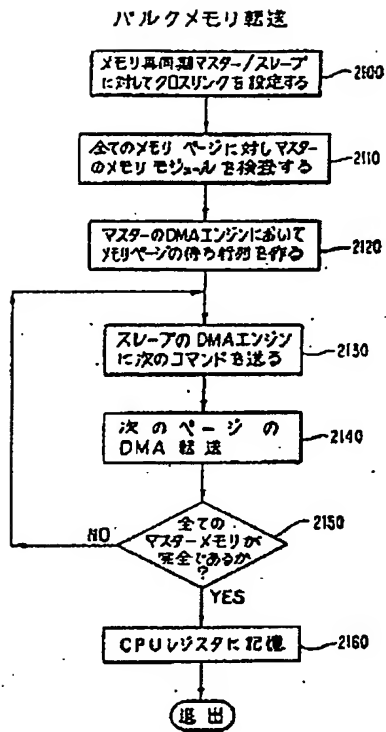
第 20E 図



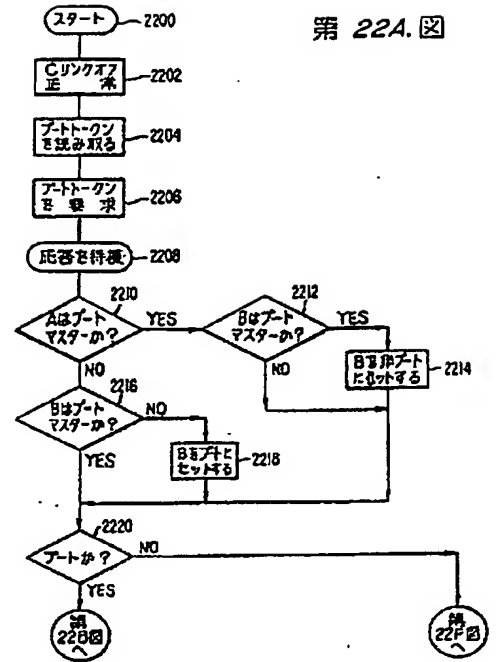
第 20C 図



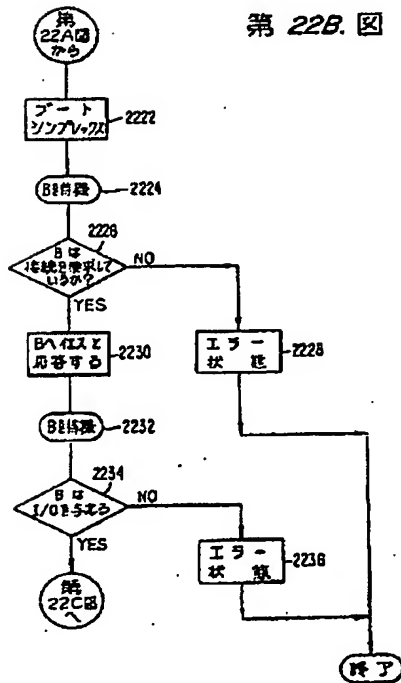
第 21. 図



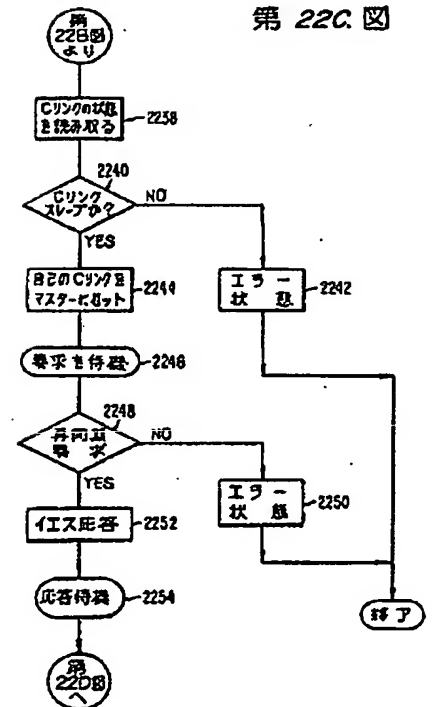
第 22A. 図



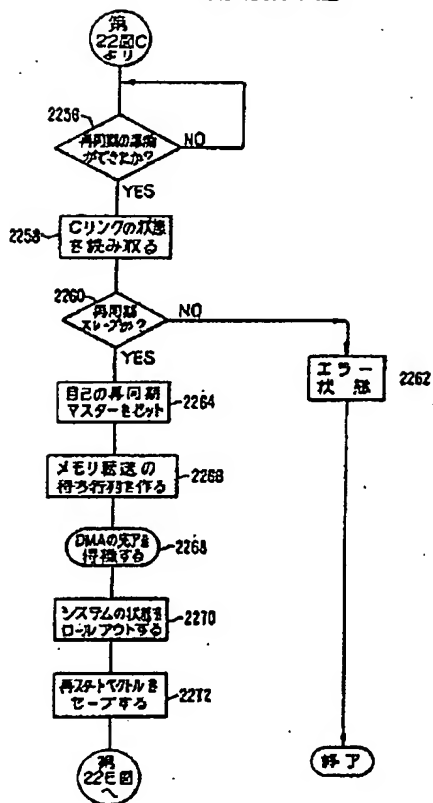
第 22B. 図



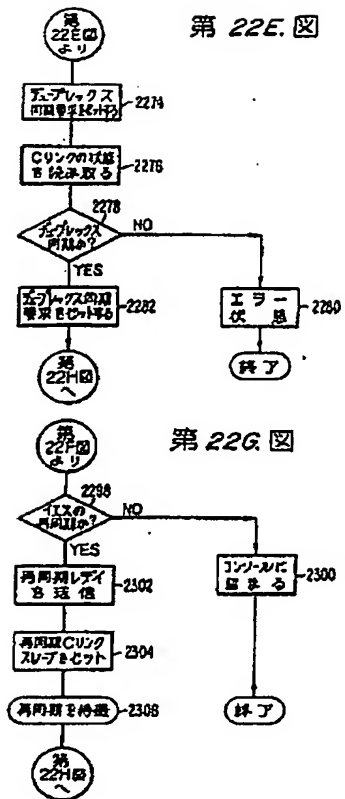
第 22C. 図



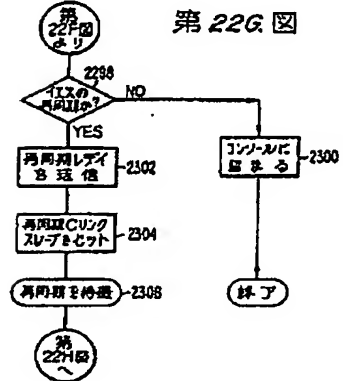
第 22D. 図



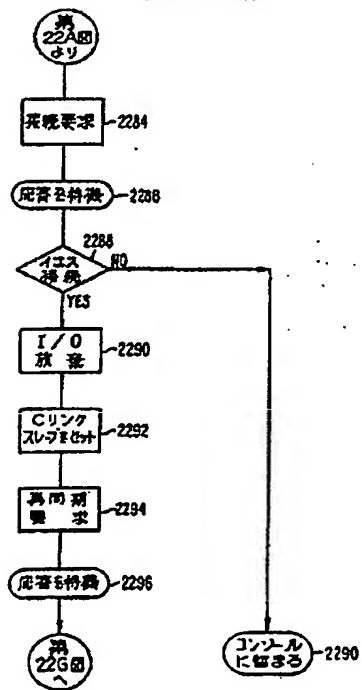
第 22E. 図



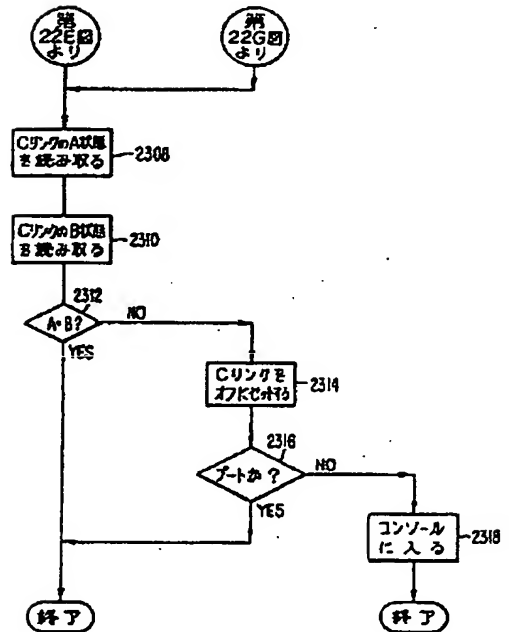
第 22G. 図



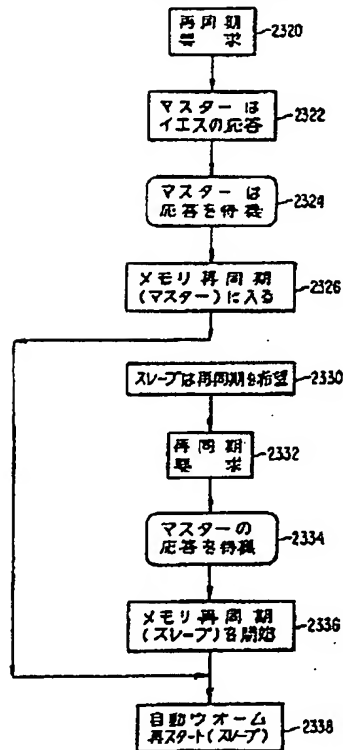
第 22F. 図



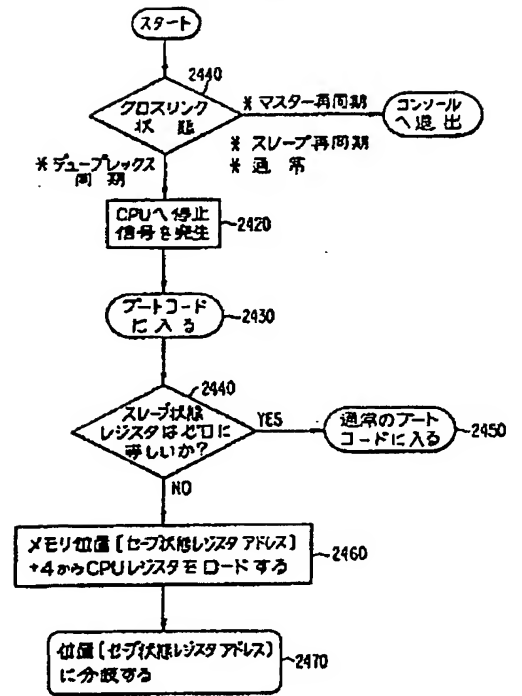
第 22H. 図



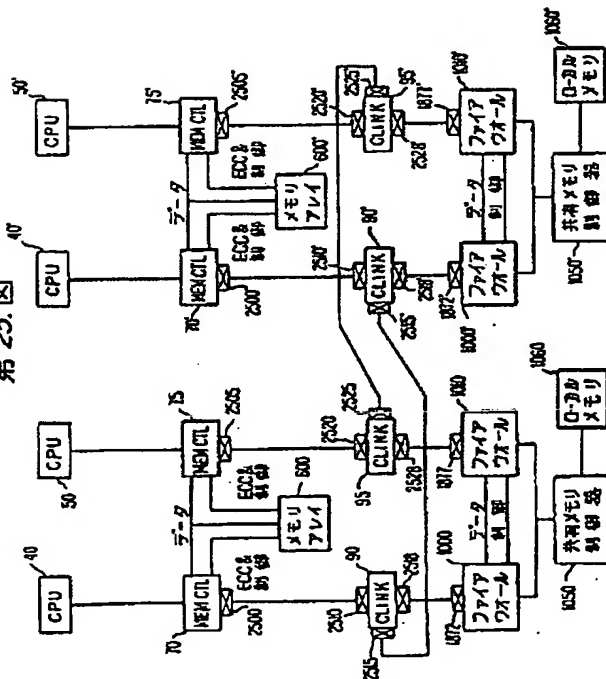
第 23. 図
メモリ再同期



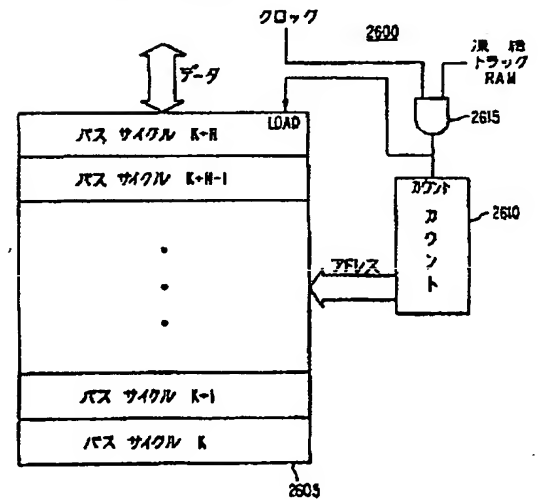
第 24. 図
ウォーム再起動



第 25. 図



第 26. 図
トレース RAM



```

graph TD
    2700[全てのトレースRAMを初期化する] --> 2702[メモリマシニング]
    2702 --> 2705{28日の入力か?}
    2705 -- YES --> 2706[内実交換経路]
    2705 -- NO --> 2710[入力プラグをセット]
    2710 --> 2711[過渡交換経路]
    2711 --> 2720[トレース RAMの内容をトレース RAMバス経路でファイアウォールに読み込む]
    2720 --> 2735[診断プロセッサがトレースデータを読み取る]
    2735 --> 2740[エラー検出器でスタート]
    2740 --> 2750{ルール同士を比較}
    2750 -- NO --> 2754{エラー発生するか?}
    2754 -- NO --> 2758{最初のトレースRAMチェックか?}
    2758 -- NO --> 2764[メモリマシニングを繰り返す]
    2758 -- YES --> 2755[予備のレベルを交換位置として記録]
    2755 --> 2762[入力プラグの取り外し]
    2762 --> 2764
    2750 -- YES --> 2752{エラーがあるか?}
    2752 -- NO --> 2754
    2752 -- YES --> 2770{動作のソースか?}
    2770 -- NO --> 2775[ソースに対してバックアップ]
    2775 --> 2770
    2770 -- YES --> 2780[動作のソースを交換位置として記録]
    2780 --> 2762
  
```

Flowchart of the program for the first embodiment:

- 2700: 全てのトレースRAMを初期化する (Initialize all trace RAM)
- 2702: メモリマシニング (Memory mapping)
- 2705: 28日の入力か? (Is it input for 28 days?)
 - YES: 内実交換経路 (Internal exchange path)
 - NO: 2710
- 2710: 入力プラグをセット (Set input plug)
- 2711: 過渡交換経路 (Transition exchange path)
- 2720: トレース RAMの内容をトレース RAMバス経路でファイアウォールに読み込む (Load trace RAM content to firewall via trace RAM bus path)
- 2735: 診断プロセッサがトレースデータを読み取る (Diagnostic processor reads trace data)
- 2740: エラー検出器でスタート (Start error detector)
- 2750: ルール同士を比較 (Compare rules)
 - NO: 2754
 - YES: 2752
- 2752: エラーがあるか? (Is there an error?)
 - NO: 2754
 - YES: 2770
- 2754: エラー発生するか? (Does error occur?)
 - NO: 2758
 - YES: 2755
- 2755: 予備のレベルを交換位置として記録 (Record as exchange position)
- 2758: 最初のトレースRAMチェックか? (Is it the first trace RAM check?)
 - NO: 2764
 - YES: 2755
- 2762: 入力プラグの取り外し (Remove input plug)
- 2764: メモリマシニングを繰り返す (Repeat memory mapping)
- 2770: 動作のソースか? (Is it the source of operation?)
 - NO: 2775
 - YES: 2780
- 2775: ソースに対してバックアップ (Backup source)
- 2780: 動作のソースを交換位置として記録 (Record as exchange position)

[illegible]

```

graph TD
    Start([開始  
28A図より]) --> 2865[直列クロスリンク  
によりエラー分  
析結果を他の  
ゾーンへ送る]
    2865 --> 2868{ゾーンは  
正常か?}
    2868 -- 応答なし --> 2870[他のゾーンが  
ブロード、欠陥を  
報告する]
    2868 -- 応答受信 --> 2877{エラー  
分析か?}
    2877 -- 別のゾーンが欠陥 --> 2879[他のゾーンを  
欠陥と報告]
    2877 -- クロスリンクが欠陥 --> 2881[直列クロスリンクにより  
欠陥のクロスリンク部を  
識別しようとする]
    2879 --> 2875[エラ  
ーは修正  
される]
    2881 --> 2883[欠陥ゾーンを修正し  
欠陥を報告する]
    2883 --> 2875
    2875 --> End([分析終了])
  
```

```

graph TD
    Start([高  
28A図  
より]) --> 2885[直列クロスリンクに  
より分析も他のゾーン  
へ送る]
    2885 --> 2887{ゾーンは  
応答するか?}
    2887 -- 応答なし --> 2889[他のゾーンを  
欠陥と報告]
    2887 -- 応答受信 --> 2893{エラー  
分析か?}
    2893 -- クロスリンク  
が欠陥 --> 2895[他のゾーンも  
クロスリンクを  
欠陥と報告]
    2893 -- エラー分析か? --> 2889
    2889 --> 2891[エラー状態を  
クリアする]
    2895 --> 2891
    2891 --> End([分析終了])
  
```

63.12.28

昭和 年 日

特許庁長官 吉 田 文 毅 殿

1. 事件の表示 昭和63年特許願第222174号

2. 発明の名称 二重ゾーンの耐欠陥コンピュータシステム

3. 補正をする者

事件との関係 出 願 人

名 称 デジタル イクイブメント
コーポレーション

4. 代 理 人

住 所 東京都千代田区丸の内3丁目3番1号
電話(代) 211-8741

氏 名 (5995) 弁理士 中 村 稔

5. 補正命令の日付 昭和63年11月29日

6. 補正の対象 願書の特許出願人の図
代理権を証明する書面
全 図 面

7. 補正の内容 別紙のとおり

願書に最初に添付した図面の浄書
(内容に変更なし)

